

SOSNet: Second Order Similarity Regularization for Local Descriptor Learning

Yurun Tian^{*1,2} Xin Yu³ Bin Fan^{†1} Fuchao Wu¹ Huub Heijnen⁴ Vassileios Balntas^{†,4}

¹National Laboratory of Pattern Recognition, Institute of Automation,
 Chinese Academy of Sciences, Beijing, China

²University of Chinese Academy of Science, Beijing, China

³Australian Center for Robotic Vision, Australian National University

⁴Scape Technologies

{yurun.tian,bfan,fcwu}@nlpr.ia.ac.cn xin.yu@anu.edu.au {huub,vassileios}@scape.io

Abstract

Despite the fact that Second Order Similarity (SOS) has been used with significant success in tasks such as graph matching and clustering, it has not been exploited for learning local descriptors. In this work, we explore the potential of SOS in the field of descriptor learning by building upon the intuition that a positive pair of matching points should exhibit similar distances with respect to other points in the embedding space. Thus, we propose a novel regularization term, named Second Order Similarity Regularization (SOSR), that follows this principle. By incorporating SOSR into training, our learned descriptor achieves state-of-the-art performance on several challenging benchmarks containing distinct tasks ranging from local patch retrieval to structure from motion. Furthermore, by designing a von Mises-Fischer distribution based evaluation method, we link the utilization of the descriptor space to the matching performance, thus demonstrating the effectiveness of our proposed SOSR. Extensive experimental results, empirical evidence and in-depth analysis are provided, indicating that SOSR can significantly boost the matching performance of the learned descriptor.

1. Introduction

The process of describing local patches is a fundamental component in many computer vision tasks such as 3D reconstruction [31, 33], large scale image localization [30] and image retrieval [29]. Early efforts mainly focused on the heuristic design of hand-crafted descriptors, by applying a set of filters to the input patches. In recent years,

^{*}Research conducted while Yurun and Xin were interns at Scape Technologies

[†]Corresponding authors

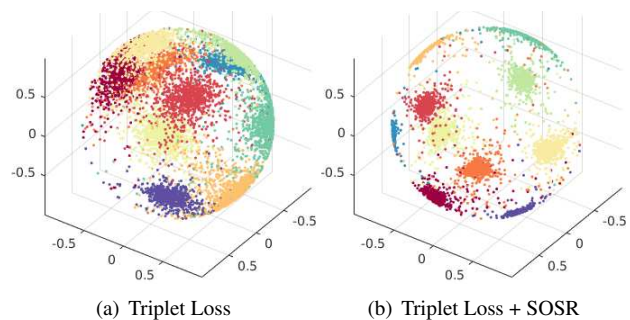


Figure 1. Qualitative results of our proposed SOSR on features learned for the 10 digits of the MNIST [19] dataset. Each digit is represented by a different colour on the unit sphere. We can observe that by using our SOSR method that encourages second order similarity, more compact individual clusters are learned compared to standard triplet loss.

large datasets with corresponding ground truths have led to the development of large scale learning methods, which stimulated a wave of works on descriptor learning. Recent work has shown that these learning based methods are able to significantly outperform their hand-crafted counterparts [1, 22].

One of the most important challenges of learning based methods is the design of suitable loss functions for the training stage. Since nearest neighbour matching is done directly using Euclidean distances, most of the recent methods focus on optimizing objectives related to First Order Similarity (FOS) by forcing descriptors from matching pairs to have smaller L_2 distance than non-matching ones [34, 2, 26, 17, 36, 15].

Second Order Similarity (SOS) has been used for graph matching and clustering tasks [10, 11, 43], due to the fact that it can capture more structural information such as shape and scale, while at the same time being robust to deforma-

tions and distortions. On the other hand, FOS and nearest neighbor matching are only limited to pairwise comparisons. However, utilizing SOS for large scale problems typically requires significant computational power [10, 11, 43], and thus matching and reconstruction tasks still rely on brute force or approximate nearest neighbor matching [31]. In this work, we explore the possibility of using SOS for learning high performance local descriptors. In particular, we are interested in formulating a SOS constraint as a regularization term during training, in order to harness its power during the matching stage, without any computational overhead.

Evaluation of descriptors is also a key issue. A good evaluation method can provide insights for designing descriptors. Performance indicators, such as false positive rate [7] and mean average precision [1], are widely used. However, it is still unclear how the utilization of the descriptor space, such as the degrees of intra-class concentration and inter-class dispersion, contributes to the final performance. Therefore, in order to explain the impact of SOS on the matching performance, we further introduce an evaluation method based on the von Mises-Fisher distribution [6].

Our main contributions are: (1) We introduce a novel regularization method, named Second Order Similarity Regularization (SOSR), that enforces Second Order Similarity (SOS) consistency. To the best of our knowledge, SOS has not been incorporated into the process of learning local feature descriptors. (2) By combining our SOSR with a triplet loss, our learned descriptor is able to significantly outperform previous ones and achieves state-of-the-art results on several benchmarks related to local descriptors. (3) We introduce a new evaluation method that is based on the von Mises-Fisher distribution for examining the utilization of the descriptor space. The proposed evaluation method can illustrate links between distributions of descriptors on the hypersphere and their matching performance.

This paper is organized as follows: In Sec. 2, we briefly review related works. In Sec. 3 and Sec. 4, we introduce our Second Order Similarity Regularization as well as a new method for evaluating the descriptors on a unit hypersphere. Subsequently, in Sec. 5, we present results on several challenging benchmarks. Lastly, we conduct ablation study of our proposed SOSR in Sec. 6.

2. Related works

Early works on local patch description focused on low level processes such as gradient filters and intensity comparisons, including SIFT [21], GLOH [24], DAISY [39], DSP-SIFT [12] and LIOP [38]. A comprehensive review can be found in [25].

With the emergence of annotated patch datasets [7], a significant amount of data-driven methods focused on improving hand-crafted representations using machine learn-

ing methods. The authors of [8, 9] use linear projections to learn discriminative descriptors, while convex optimization is used for learning an optimal descriptor sampling configuration in [35]. BinBoost [37] is trained based on a boosting framework, and in RFD [13], the most discriminative receptive fields are learned based on the labeled training data. BOLD [3] uses patch specific adaptive online selection of binary intensity tests.

Convolutional neural networks (CNNs) enable end-to-end descriptor learning from raw local patches, and have become the de-facto standard for learning local patch descriptors in recent years. MatchNet [14] adopts a Siamese network for local patch matching, while DeepCompare [40] further explores various network architectures. Song *et al.* [27] propose the lifted structured embedding for the task of feature embedding. DeepDesc [34] removes the need for a specially learned distance metric layer, and instead uses Euclidean distances and hard sample mining. TFeat [2] uses triplet learning constraints with shallow convolutional networks and fast hard negative mining, and L2Net [36] applies progressive sampling with a loss function that takes into account the whole training batch while producing descriptors that are normalized to unit norm. The L2Net architecture was widely adopted by consequent works. HardNet [26] surpasses L2Net by implementing a simple hinge triplet loss with the “hardest-within-batch” mining, confirming the importance of the mining strategy. Keller *et al.* [17] propose to learn consistently scaled descriptors by a mixed context losses and a scale aware sampling. Instead of focusing on patch matching, DOAP [15] imposes a retrieval based ranking loss and achieves the current state of the art performance on several benchmarks. GeoDesc [22] integrates geometry constraints from multi-view reconstructions to benefit the learning process by improving the training data. The authors of [41], propose a Global Orthogonal Regularization term to better exploit the unit hypersphere.

While the recent improvements in the field of learning CNN patch descriptors are significant, methods mentioned above are limited to optimizing the FOS measured by L_2 distances of positive and negative pairs, and the potential of SOS has not been exploited in the area of descriptor learning. On the other hand, graph matching algorithms [10, 11, 43] have been developed based on SOS due to its robustness to shape distortions. Furthermore, [20] proves that using SOS can achieve better cluster performance. Thus, our key idea is to introduce second order similarity constraints at the training stage for robust patch description.

3. Learning Descriptor with Second Order Similarities

In this section, we introduce how to incorporate SOS as a regularization term into our training procedure. By em-

playing FOS and SOS losses, we train our network in an end-to-end manner.

3.1. Preliminaries

For a training batch consisting of N pairs of matching patches, a convolutional neural network is applied to each patch so as to extract its descriptor. The corresponding positive descriptor pairs are denoted as $\{\mathbf{x}_i, \mathbf{x}_i^+\}_{i=1\dots N}$

3.2. First Order Similarity Loss

First Order Similarity (FOS) loss, which enforces distances between matching descriptors to be small while those of non-matching ones to be large, has been widely used for learning local descriptors [2, 36, 15, 22]. In our method, we first employ a loss term to constraint the FOS as follows:

$$\mathcal{L}_{FOS} = \frac{1}{N} \sum_{i=1}^N \max(0, t + d_i^{\text{pos}} - d_i^{\text{neg}})^2, \quad (1)$$

$$d_i^{\text{pos}} = d(\mathbf{x}_i, \mathbf{x}_i^+),$$

$$d_i^{\text{neg}} = \min_{\forall j, j \neq i} (d(\mathbf{x}_i, \mathbf{x}_j), d(\mathbf{x}_i, \mathbf{x}_j^+), d(\mathbf{x}_i^+, \mathbf{x}_j), d(\mathbf{x}_i^+, \mathbf{x}_j^+)),$$

where t is the margin, $d(\mathbf{u}, \mathbf{v}) = \|\mathbf{u} - \mathbf{v}\|_2$ is the L_2 distance, d_i^{pos} indicates the distance between a positive pair and d_i^{neg} represents the distance between a negative pair. We employ the same mining strategy as in HardNet [26] to find the ‘‘hardest-within-batch’’ negatives. Note that, in Eqn. (1), we use a Quadratic Hinge Triplet (QHT) loss instead of the conventional Hinge Triplet (HT) Loss. Compared with HT, QHT weights the gradients with respect to the parameters of the network by the magnitude of the loss. This means that the larger $d_i^{\text{neg}} - d_i^{\text{pos}}$ is, the smaller the gradients are. In Sec. 6.1 we provide evidence that this simple modification can lead to significant performance improvements.

3.3. Second Order Similarity Regularization

Besides the first order constraints imposed by \mathcal{L}_{FOS} , it has been demonstrated that incorporating information from higher order similarities can improve the performance of clustering [20] and graph matching [10]. Thus, we propose to impose a second order constraint to further supervise the process of descriptor learning.

A training mini-batch can be viewed as two sets of descriptors with one-to-one correspondence, *i.e.*, $\{\mathbf{x}_i\}_{i=1\dots N}$ and $\{\mathbf{x}_i^+\}_{i=1\dots N}$. For this case, we define the second order similarity between \mathbf{x}_i and \mathbf{x}_i^+ as:

$$d^{(2)}(\mathbf{x}_i, \mathbf{x}_i^+) = \sqrt{\sum_{j \neq i}^N (d(\mathbf{x}_i, \mathbf{x}_j) - d(\mathbf{x}_i^+, \mathbf{x}_j^+))^2}, \quad (2)$$

where $d^{(2)}(\mathbf{x}_i, \mathbf{x}_i^+)$ measures the similarity between \mathbf{x}_i and \mathbf{x}_i^+ from the perspectives of $\{\mathbf{x}_j\}_{j \neq i}$ and $\{\mathbf{x}_j^+\}_{j \neq i}$, using the differences between distances.

In order to enforce SOS, we formulate our SOSR regularization term as:

$$\mathcal{R}_{SOS} = \frac{1}{N} \sum_{i=1}^N d^{(2)}(\mathbf{x}_i, \mathbf{x}_i^+). \quad (3)$$

Note that \mathcal{R}_{SOS} does not force distances between matching descriptors to decrease or distances between non-matching ones to increase. Thus, it cannot be solely used without an \mathcal{L}_{FOS} term, and can only be served as a regularization term.

3.4. Objective Function For Training

Our goal is to learn a robust descriptor in terms of both FOS and SOS, therefore, our total objective function is expressed as:

$$\mathcal{L}_T = \mathcal{L}_{FOS} + \mathcal{R}_{SOS}, \quad (4)$$

where the two terms are weighted equally.

3.5. Implementation Details

During training, we observed that using all the samples in a mini-batch as input to the \mathcal{R}_{SOS} term led to inferior results. This is due to the fact that for a given pair of matching descriptors, many of their non-matching descriptors are already far away. Thus, these distant negatives need no further optimization. Subsequently, SOS calculated on these ‘‘easy’’ negatives may produce noisy gradients and therefore damage the performance. Inspired by the concept of active graph in [11], we employ nearest neighbor search to exclude those far away negatives for each positive pair. Let z_i be the class label of the i^{th} positive pair, and \mathbf{c}_i be the i^{th} set of class labels. In particular, \mathbf{c}_i stores the class labels which are within the K Nearest Neighbors (K NN) of the i^{th} positive pair. Thus, we define the criterion of neighbor selection for each \mathbf{c}_i as:

$$\mathbf{c}_i = \{z_j : \mathbf{x}_i \in K\text{NN}(\mathbf{x}_j) \vee \mathbf{x}_i^+ \in K\text{NN}(\mathbf{x}_j^+)\}, \quad (5)$$

$$\forall j \in 1 \dots N, j \neq i$$

where $K\text{NN}(\mathbf{x}_i)$ denotes the K Nearest Neighbors of descriptor \mathbf{x}_i . Note that there is a possibility of intersection between the $K\text{NN}(\mathbf{x}_i)$ and $K\text{NN}(\mathbf{x}_i^+)$ sets. Thus, the cardinality of \mathbf{c}_i ranges from K to $2K$. Therefore, in Eqn (3) we calculate SOS for the i^{th} pair as:

$$d^{(2)}(\mathbf{x}_i, \mathbf{x}_i^+) = \sqrt{\sum_{j \neq i, z_j \in \mathbf{c}_i}^N (d(\mathbf{x}_i, \mathbf{x}_j) - d(\mathbf{x}_i^+, \mathbf{x}_j^+))^2}. \quad (6)$$

We adopt the architecture of L2Net [36] to embed local patches to 128-dimensional descriptors. Note that all

descriptors are normalized to unit vectors. To prevent overfitting, we also employ a dropout layer with a drop rate of 0.1 before the last convolutional layer. Similar to previous works [36, 26], all patches are resized to 32×32 and normalized by subtracting the per-patch mean and dividing the per-patch standard deviation. We use the PyTorch library [28] to train our local descriptor network. Our network is trained for 100 epochs using the Adam optimizer [18] with $\alpha = 0.01$, $\beta_1 = 0.9$ and $\beta_2 = 0.999$ as in the default settings. For the training hyperparameters, the number of training pairs N is set to 512, *i.e.*, the batch size is 1024, K is set to 8, that is, 8 nearest neighboring pairs are selected to calculate SOS for a given pair, and the margin t in the FOS loss is set to be 1.

4. Evaluating the Unit Hypersphere Utilization

Indicators like false positive rate and mean average precision have been widely used for evaluating the performance of descriptors [1, 39]. However, such indicators fail to provide insights on properties of the learned descriptors, *i.e.*, how the utilization of the descriptor space such as the intra-class and inter-class distributions, contribute to the final performance. To investigate this, previous works [41, 36] visualize the distributions of the positive and negative distances as histograms. However, while such visualizations illuminate the distance distributions, they fail to capture the structure of the learned descriptor space.

Since most modern methods rely on normalized descriptors, we propose to leverage the von Mises-Fisher (vMF) distribution which deals with the statistical properties of unit vectors that lie on hyperspheres (interested readers can find more information in [6]). A q -dimensional descriptor, can be thought as a random point on the $(q-1)$ -dimensional unit hypersphere \mathbb{S}^{q-1} . Specifically, a random unit vector \mathbf{x} (*i.e.*, $\|\mathbf{x}\|_2 = 1$) obeys a q -variate vMF distribution if its probability density function is as follows:

$$f(\mathbf{x}|\boldsymbol{\mu}, \kappa) = c_q(\kappa)e^{\kappa\boldsymbol{\mu}^T\mathbf{x}}, \quad (7)$$

where $\|\boldsymbol{\mu}\|_2 = 1$, $q \geq 2$ and $\kappa \geq 0$. The normalizing constant $c_q(\kappa)$ is defined as:

$$c_q(\kappa) = \frac{\kappa^{q/2-1}}{(2\pi)^{q/2}I_{q/2-1}(\kappa)}, \quad (8)$$

where $I_k(\cdot)$ is the modified Bessel function of the first kind and order k . The vMF density $f(\mathbf{x}|\boldsymbol{\mu}, \kappa)$ is parameterized by the mean direction $\boldsymbol{\mu}$ and the concentration parameter κ . κ is used to characterize how strongly the unit vectors drawn from $f(\mathbf{x}|\boldsymbol{\mu}, \kappa)$ are concentrated in the mean direction $\boldsymbol{\mu}$, with larger values of κ indicating stronger concentration. In particular, when $\kappa = 0$, $f(\mathbf{x}|\boldsymbol{\mu}, \kappa)$ reduces to the uniform distribution on \mathbb{S}^{q-1} , and as $\kappa \rightarrow \infty$, $f(\mathbf{x}|\boldsymbol{\mu}, \kappa)$ approaches a point density.

According to [6], the maximum likelihood estimation of κ can be obtained from the following equation:

$$A(\hat{\kappa}) = \frac{I_{q/2}(\hat{\kappa})}{I_{q/2-1}(\hat{\kappa})} = \bar{R} = \frac{1}{N} \left\| \sum_{i=1}^N \mathbf{x}_i \right\|_2, \quad (9)$$

where $\hat{\kappa}$ is the estimation of κ and \bar{R} is called the mean resultant length. Since $A(\cdot)$ is a ratio of Bessel functions [4] with no analytic inverse, we cannot directly take $\hat{\kappa} = A^{-1}(\bar{R})$. According to [5], $\hat{\kappa}$ can be approximated by a monotonically increasing function of \bar{R} , where $\bar{R} = 0$ leads to $\hat{\kappa} = 0$ and $\bar{R} = 1$ indicates $\hat{\kappa} = \infty$. Therefore, \bar{R} can be used as a proxy for measuring κ .

Descriptors from the i^{th} class can be interpreted as samples drawn from a vMF distribution $f_{\text{intra}}^i(\mathbf{x}|\boldsymbol{\mu}_i, \kappa_{\text{intra}}^i)$. The cluster center $\boldsymbol{\mu}_i$ is a sample from vMF density $f_{\text{inter}}(\boldsymbol{\mu}|\boldsymbol{\nu}, \kappa_{\text{inter}})$. Further, to investigate the utilization of the unit hypersphere, we define the following parameters:

$$\begin{aligned} R_{\text{intra}} &= \frac{1}{M} \sum_i^M \bar{R}_i^{\text{intra}}, \\ R_{\text{inter}} &= \frac{1}{M} \left\| \sum_{i=1}^M \boldsymbol{\mu}_i \right\|_2, \\ \rho &= \frac{R_{\text{inter}}}{R_{\text{intra}}}, \end{aligned} \quad (10)$$

where M is the total number of classes and \bar{R}_i^{intra} is the mean resultant length for the i^{th} class. In Eqn. (10), R_{intra} and R_{inter} measure the intra-class concentration, inter-class dispersion respectively, and the ratio ρ is a overall evaluation.

The vMF distribution has been used in image clustering [4] and classification [42]. However, we propose to use it solely for evaluating the utilization of the descriptor space, since unlike in the classification tasks, current local patch datasets can not guarantee sufficient intra-class samples for accurate estimation of the vMF parameters, *e.g.*, some classes in the widely used UBC Phototour [7] dataset only have 2 samples, and such estimation errors in training stage may lead to inferior performance.

5. Experiments

We name our learned descriptor *Second Order Similarity Network* (SOSNet). In this section, we compare our SOSNet with several state-of-the-art methods, *i.e.*, DeepDesc(DDesc) [34], TFeat [2], L2Net [36], HardNet(HNet) [26], HardNet with GOR [41], Scale-Aware Descriptor [17], DOAP [15] and GeoDesc [22]. We perform our experiments on three publicly available datasets, namely UBC Phototour [7], HPatches [1] and ETH SfM[32]. For TFeat [2], L2Net [36], HardNet [26], and

Train	Notredame	Yosemite	Liberty	Yosemite	Liberty	Notredame	Mean
Test	Liberty		Notredame		Yosemite		
SIFT [21]	29.84		22.53		27.29		26.55
DeepDesc [34]	10.9		4.40		5.69		6.99
MatchNet [2]	7.04	11.47	3.82	5.65	11.6	8.70	8.05
L2Net [36]	3.64	5.29	1.15	1.62	4.43	3.30	3.24
CS L2Net [36]	2.55	4.24	0.87	1.39	3.81	2.84	2.61
HardNet [26]	1.47	2.67	0.62	0.88	2.14	1.65	1.57
HardNet-GOR [26, 41]	1.72	2.89	0.63	0.91	2.10	1.59	1.64
Michel <i>et al.</i> [17]	1.79	2.96	0.68	1.02	2.51	1.64	1.77
SOSNet	1.25	2.84	0.58	0.87	1.95	1.25	1.46
TFeat+ [2]	7.39	10.13	3.06	3.80	8.06	7.24	6.64
L2Net+ [36]	2.36	4.70	0.72	1.29	2.57	1.71	2.23
CS L2Net+ [36]	1.71	3.87	0.56	1.09	2.07	1.3	1.76
HardNet+ [26]	1.49	2.51	0.53	0.78	1.96	1.84	1.51
HardNet-GOR+ [26, 41]	1.48	2.43	0.51	0.78	1.76	1.53	1.41
DOAP+ [15]	1.54	2.62	0.43	0.87	2.00	1.21	1.45
DOAP-ST+ [15] [16]	1.47	2.29	0.39	0.78	1.98	1.35	1.38
SOSNet+	1.08	2.12	0.35	0.67	1.03	0.95	1.03
GeoDesc+ [22]	5.47		1.94		4.72		4.05
SOSNet-HP+	2.10		0.79		1.39		1.42

Table 1. Patch verification performance on the UBC phototour dataset. Numbers denote false positive rates at 95% recall. All descriptors are 128-dimensional, except that TFeat is 256. and suffix “+” indicates data augmentation. We can observe that our SOSNet outperforms other methods in all cases.

GeoDesc* [22], we use the pre-trained models released by the authors, and for GOR [41], we employ the code provided by the authors. For Scale-Aware Descriptors [17] and DOAP [15], we report their results from their published papers since their training codes and pre-trained models are unavailable.

5.1. UBC Phototour

UBC Phototour dataset [7] is currently the most widely used dataset for local patch descriptor learning. It consists of three subsets, Liberty, Notredame, and Yosemite. For evaluations on this dataset, models are trained on one subset and tested on the other two. We follow the standard evaluation protocol of [7] by using the 100K pairs provided by the authors and report the false positive rate at 95% recall.

In Table. 1, SIFT represents a baseline hand-crafted descriptor, and the others are CNN based methods. As indicated by Table. 1, our SOSNet achieves the best performance by a significant margin in comparison to the state-of-the-art approaches. Note that, DOAP incorporates a Spatial Transformer Network (STN) [16] into the network to resist geometrical distortions in the patches. In contrast, our SOSNet does not require any extra geometry rectifying layer and yet achieves superior performance. We can expect the performance of our method to further increase, by

*The training dataset of GeoDesc is not publicly available. Therefore, comparisons of GeoDesc with other methods may be unfair.

incorporating an STN. GeoDesc generates inferior results due to the possible differences between its training dataset and UBC Phototour. In addition, it is worth noting that even when trained on HPatches (SOSNet-HP+), our descriptor is able to closely match the best performing methods, which is significant since UBC and HPatches exhibit vastly different patch distributions. This is a testament to the generalization ability of our method.

5.2. HPatches

HPatches dataset [1] consists of over 1.5 million patches extracted from 116 viewpoint and illumination changing scenes. According to the different levels of geometric noise, the extracted patches can be divided into three groups: *easy*, *hard*, and *tough*. There are three evaluation tasks, patch verification, patch retrieval, and image matching. We show results for all three tasks in Fig. 2. As shown in Fig. 2, our SOSNet outperforms state-of-the-art methods on all the three tasks both for methods trained on Liberty (-LIB) and on HPatches (-HP). It is worth noting that our descriptor outperforms DOAP in the retrieval task, even though DOAP employs a ranking loss specifically designed for maximizing the mean Average Precision (mAP) for patch retrieval. This indicates that our SOSR can lead to more discriminative descriptors, without the need for a specialized ranking loss.

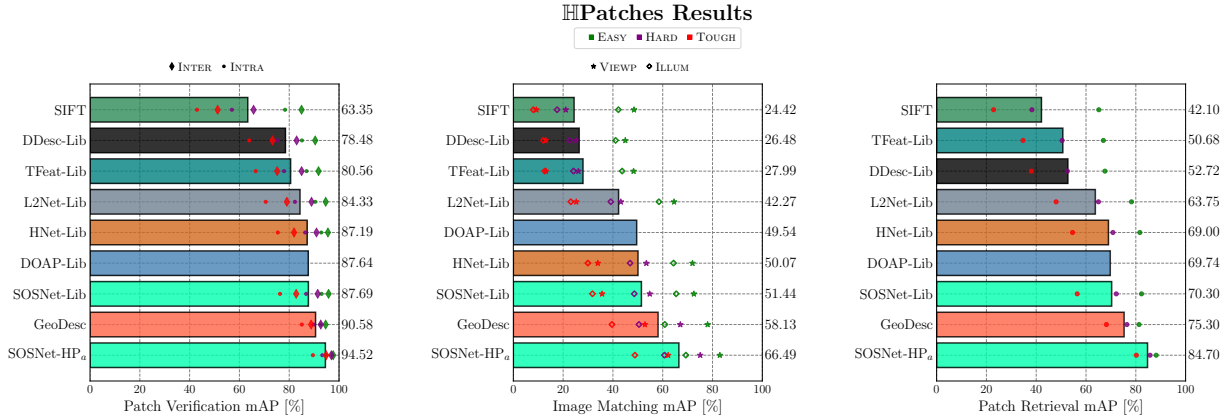


Figure 2. Verification, matching and retrieval results on test set split ‘a’ of HPatches [1]. Colour of the marker indicates EASY, HARD, and TOUGH noise. The type of the marker corresponds to the variants of the experimental settings.

5.3. ETH dataset

Different from the aforementioned datasets that focus on patches, the ETH SfM benchmark [32] aims to evaluate descriptors for a Structure from Motion (SfM) task. This benchmark investigates how different methods perform in terms of building a 3D model from a set of available 2D images.

In this experiment, we compare our SOSNet with the state-of-the-art methods by quantifying the SfM quality, *i.e.*, measuring the number of registered images, reconstructed sparse points, image observations, mean track length and mean reprojection error. Following the protocols in [32], we do not conduct the ratio test, in order to investigate the direct matching performance of the descriptors.

Table 2 shows the evaluation results of the 3D reconstruction, in which SOSNet exhibits the best overall performance. In particular, SOSNet is able to significantly outperform other methods in terms of metrics related to the density of the reconstructed 3D model, *i.e.* the number of registered sparse points, and the number of observations. It is worth noting that SOSNet produces even more matches than GeoDesc [22], which is specifically designed and trained for a SfM task. Similarly to the observations in [32, 22], SIFT achieves the smallest reprojection error on all tests, thus demonstrating that it is still an attractive choice for image matching. This can be explained by the fact that fewer matches seem to lead to a trend for lower reprojection errors. Furthermore, since the reprojection errors are less than 1px for all descriptors, we can conclude that this metric may not reflect performance differences between descriptors in practice. Finally, we can observe that our method is able to register significantly more images compared to SIFT. For example, in Madrid Metropolis sequence, SIFT was able to register only 38% of the available 2D images for

the final 3D model, while our method registered 65% of the images. This indicates that our method is more suitable for large scale and challenging reconstructions.

6. Discussion

In this section, we perform several experiments to provide a more in-depth analysis about how each component in SOSNet contributes to its final performance. Besides reporting matching performance in terms of FPR@95 rate and mAP, we also demonstrate how the proposed SOSR and other existing methods impact on the structure of the learned descriptor space, using the methodology introduced in Sec. 4.

6.1. Analysis of Performance Improvements

We argue that the performance increase of SOSNet comes from three aspects: 1) the optimization method that we employ, 2) the QHT, and 3) the proposed SOSR.

First, we investigate the impact of different optimization methods, where the two most widely adopted methods, *i.e.*, Stochastic Gradient Descent (SGD) and Adam [18] are compared. For SGD, we use a starting learning rate of 0.01 and divided it by 10 at epoch 50, and for Adam, we use the settings described in Sec. 5. As visible in Fig. 3(a) and Fig. 3(b), Adam [18] leads to better performance compared to SGD. Note that, using Hinge Triplet (HT) loss with Adam already surpasses the previous state-of-art method, *i.e.*, DOAP [15] that uses a sophisticated ranking loss.

Second, we compare QHT against HT. As shown in Fig. 3(a) and Fig. 3(b), performance improvements from HT to QHT are quite obvious for both SGD and Adam cases. This is mainly due to the fact that QHT loss adaptively weights the gradients by the magnitude of the loss, *i.e.*, $d_{\text{neg}} - d_{\text{pos}}$.

Third, we compare our SOSR with another regulariza-

		# Image	# Registered	# Sparse Points	# Observations	Track Length	Reproj. Error
Fountain	SIFT	11	11	14K	70K	4.79	0.39px
	DSP-SIFT		11	14K	71K	4.78	0.37px
	L2Net		11	17K	83K	4.88	0.47px
	GeoDesc		11	16K	83K	5.00	0.47px
	SOSNet		11	17K	85K	4.92	0.43px
Herzjesu	SIFT	8	8	7.5K	31K	4.22	0.43px
	DSP-SIFT		8	7.7K	32K	4.22	0.45px
	L2Net		8	9.5K	40K	4.24	0.51px
	Geodesc		8	9.2K	40K	4.35	0.51px
	SOSNet		8	9.7K	41K	4.26	0.53px
South Building	SIFT	128	128	108K	653K	6.04	0.54px
	DSP-SIFT		128	112K	666K	5.91	0.58px
	L2Net		128	170K	863K	5.07	0.63px
	GeoDesc		128	170K	887K	5.21	0.64px
	SOSNet		128	178K	913K	5.11	0.67px
Madrid Metropolis	SIFT	1344	500	116K	733K	6.32	0.60px
	DSP-SIFT		467	99K	649K	6.52	0.66px
	L2Net		692	254k	1067K	4.20	0.69px
	GeoDesc		809	306K	1200K	3.91	0.66px
	SOSNet		844	335K	1411K	4.21	0.70px
Gendarmenmarkt	SIFT	1463	1035	338K	1872K	5.523	0.69px
	DSP-SIFT		979	293K	1577K	5.381	0.74px
	L2Net		1168	667k	2611K	3.91	0.73px
	GeoDesc		1208	779K	2903K	3.72	0.74px
	SOSNet		1201	816K	3255K	3.984	0.77px

Table 2. Evaluation results on ETH dataset [32] for SfM. We can observe that our proposed SOSNet significantly outperforms other methods in terms of the number of registered sparse points and number of observations. This indicates that the models that are built using our descriptor are significantly denser.

tion term recently proposed in [41], *i.e.*, the Global Orthogonal Regularization (GOR). As shown in Fig. 3(a) and Fig. 3(b), SOSR achieves significant and consistent performance improvements across all training epochs, while the FPR curves with and without GOR are sometimes intertwined, showing minor performance enhancement, and this phenomenon is also observed in [15].

To sum up, Adam, QHT and SOSR bring on average 11.63%, 5.46%, and 19.49% relative performance improvements, respectively. Note that when calculating the relative performance increase caused by SOSR, we average the FPR@95 over HT, QHT for both SGD and Adam from epoch 50 to epoch 100, with the same rule applying to Adam and QHT.

6.2. Impact of K and N

As described in Sec. 4, each training batch is formed by N pairs of patches, and within each batch, K nearest neighbors are used to calculate SOSR. In this section, we analyze the impact of the hyperparameters N and K on the matching performance SOSNet. Specifically, we vary N and K from 256 to 2048 and 4 to 32, respectively. All models are trained on Liberty and tested on the other two subsets, *i.e.*, Notredame and Yosemite. We report the mean FPR@95 of the two test sets in Fig. 3(c). Across all the settings, $N = 512$ with $K = 8$ achieves the best performance.

6.3. Analysis of the Descriptor Space

To visualize the changes in the descriptor space caused by SOSR, we first conduct a toy experiment on the MNIST [19] dataset. Specifically, we modify the L2Net architecture [36] by setting the number of output channels of the last convolutional layer as 3. The network is trained with a batch size of 20, *i.e.*, each batch contains 10 classes with 2 images per class. After training, we visualize the distribution of the descriptors on a unit sphere in Fig 1. It can be clearly seen that SOSR makes each cluster more concentrated, thus indicating that in the low dimensional space enforcing SOS constraint improves FOS.

Unlike clustering 10 classes of images on a unit sphere, it is hard to directly visualize the distribution of descriptors on \mathbb{S}^{127} from tens of thousands of classes. We have tried dimensionality reduction techniques such as tSNE [23]. However, it is hard to get any insightful conclusions visually about the structure of the descriptor space due to the distortions introduced by the dimensionality reduction process.

In order to provide quantitative results, we employ the evaluation method described in Sec. 4. Specifically, we evaluate Eqn. (10) by using 90K randomly selected classes from the HPatches dataset. To avoid noisy estimation of R_{inter} , like R_{intra} , we compute it by averaging 10K random tests, where in the i^{th} test a R_i^{inter} is estimated by sampling descriptors randomly from all classes (one descriptor per class). The results are shown in Fig. 4, and several in-

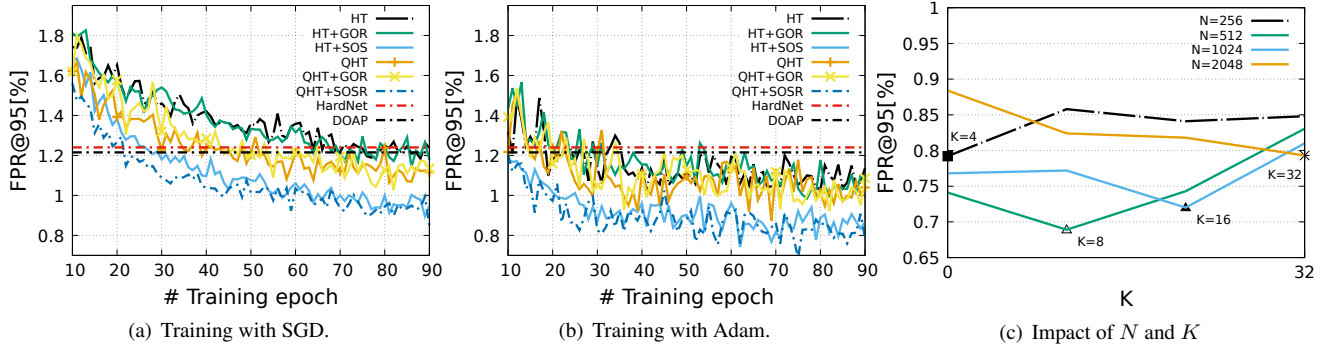


Figure 3. Analysis of Performance Improvements. HT stands for Hinge Triplet and QHT stands for Quadratic Hinge Triplet. All models are trained on Liberty and the FPR@95 is averaged over Yosemite and Notredame. Note that SOSNet is denoted as QHT+SOSR.

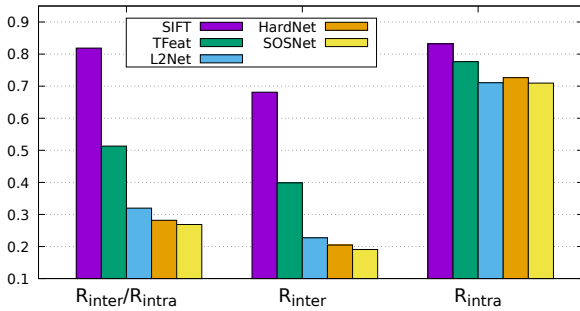


Figure 4. Performance in terms of the mean resultant length for the HPatches dataset. SIFT [21] is normalized to have unit length.

interesting observations can be drawn:

- The ratio ρ drops in accordance with the performance ranking, *i.e.*, $SIFT < TFeat < L2Net < HardNet < SOSNet$, indicating that it is a reasonable performance indicator.
- As performance increases from SIFT to SOSNet, R_{inter} decreases monotonically, showing that the more space on the hypersphere has been exploited. Specifically, descriptors can harness more expressive power of \mathbb{S}^{127} by exploiting more space of it, thus leading to better matching performance.
- With more space on \mathbb{S}^{127} being exploited, R_{intra} also drops, which means there are more scattered intra-class distributions. However, as long as the inter-class distribution is scattered enough, less concentrated intra-class distributions do not damage the matching performance.
- SIFT has the highest R_{intra} , indicating that the intra-class distributions are very concentrated. Meanwhile, it also has the highest R_{inter} , showing that most of the classes are gathered in a small region of \mathbb{S}^{127} , while leaving most of the area unexploited.

It is interesting to note that in low dimensional space (Fig. 1), SOSR helps to make more concentrated intra-class distributions, while in high dimensional space (Fig. 4) it helps to make inter-class distribution more scattered. We argue that this phenomenon is related to the dimension of the descriptors. When the descriptor space is of less complexity, *e.g.*, \mathbb{S}^2 , there is less flexibility for adjusting descriptor distributions. Therefore to ensure high second order similarity, SOSR enforces descriptors from the same class to become one point. In contrast, for high dimensional descriptor space which is hard to visualize or even imagine, *e.g.*, \mathbb{S}^{127} , experimental results show that SOSR leads to a more scatter inter-class distribution, *i.e.*, the descriptors exploit more area on the hypersphere. To sum up, the adjustment of the descriptor space by SOSR leads to better matching performance, thus demonstrating our intuition that enforcing SOS in the training stage is reasonable.

7. Conclusions

In this work, we propose a regularization term named Second Order Similarity Regularization (SOSR) for the purpose of incorporating second order similarities into the learning of local descriptors. We achieve state-of-the-art performance on several standard benchmarks on different tasks including patch matching, verification, retrieval, and 3D reconstruction, thus demonstrating the effectiveness of the proposed SOSR. Furthermore, we propose an evaluation method based on the von Mises-Fisher distribution to investigate the impact of enforcing second order similarity during training. By leveraging this evaluation method, we observe how the intra-class and inter-class distributions affect the performance of different descriptors.

Acknowledgement. This work is supported by the National Natural Science Foundation of China (61573352,61876180), the Young Elite Scientists Sponsorship Program by CAST (2018QNRC001), the Australian Research Council Centre of Excellence for Robotic Vision (project number CE140100016) and Scape Technologies.

References

- [1] Vassileios Balntas, Karel Lenc, Andrea Vedaldi, and Krystian Mikolajczyk. Hpatches: A benchmark and evaluation of handcrafted and learned local descriptors. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, volume 4, page 6, 2017. 1, 2, 4, 5, 6
- [2] Vassileios Balntas, Edgar Riba, Daniel Ponsa, and Krystian Mikolajczyk. Learning local feature descriptors with triplets and shallow convolutional neural networks. In *British Machine Vision Conference (BMVC)*, volume 1, page 3, 2016. 1, 2, 3, 4, 5
- [3] Vassileios Balntas, Lilian Tang, and Krystian Mikolajczyk. Bold-binary online learned descriptor for efficient image matching. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2367–2375, 2015. 2
- [4] Arindam Banerjee, Inderjit S Dhillon, Joydeep Ghosh, and Suvrit Sra. Clustering on the unit hypersphere using von mises-fisher distributions. *Journal of Machine Learning Research*, 6(Sep):1345–1382, 2005. 4
- [5] Arindam Banerjee, Inderjit S Dhillon, Joydeep Ghosh, and Suvrit Sra. Clustering on the unit hypersphere using von mises-fisher distributions. *Journal of Machine Learning Research*, 6(Sep):1345–1382, 2005. 4
- [6] Edward Batschelet. *Circular statistics in biology*, volume 111. Academic press London, 1981. 2, 4
- [7] Matthew Brown, Gang Hua, and Simon Winder. Discriminative learning of local image descriptors. *IEEE PAMI*, 33(1):43–57, 2011. 2, 4, 5
- [8] Matthew Brown, Gang Hua, and Simon Winder. Discriminative learning of local image descriptors. *IEEE PAMI*, 33(1):43–57, 2011. 2
- [9] Hongping Cai, Krystian Mikolajczyk, and Jiri Matas. Learning linear discriminant projections for dimensionality reduction of image descriptors. *IEEE PAMI*, 33(2):338–352, 2011. 2
- [10] Minsu Cho, Jungmin Lee, and Kyoung Mu Lee. Reweighted random walks for graph matching. In *European Conference on Computer Vision (ECCV)*, pages 492–505. Springer, 2010. 1, 2, 3
- [11] Minsu Cho and Kyoung Mu Lee. Progressive graph matching: Making a move of graphs via probabilistic voting. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 398–405. IEEE, 2012. 1, 2, 3
- [12] Jingming Dong and Stefano Soatto. Domain-size pooling in local descriptors: Dsp-sift. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5097–5106, 2015. 2
- [13] Bin Fan, Qingqun Kong, Tomasz Trzcinski, Zhiheng Wang, Chunhong Pan, and Pascal Fua. Receptive fields selection for binary feature description. *IEEE Transactions on Image Processing*, 23(6):2583–2595, 2014. 2
- [14] Xufeng Han, Thomas Leung, Yangqing Jia, Rahul Sukthankar, and Alexander C Berg. Matchnet: Unifying feature and metric learning for patch-based matching. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3279–3286, 2015. 2
- [15] Kun He, Yan Lu, and Stan Sclaroff. Local descriptors optimized for average precision. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 596–605, 2018. 1, 2, 3, 4, 5, 6, 7
- [16] Max Jaderberg, Karen Simonyan, Andrew Zisserman, et al. Spatial transformer networks. In *Advances in Neural Information Processing Systems (NIPS)*, pages 2017–2025, 2015. 5
- [17] Michel Keller, Zetao Chen, Fabiola Maffra, Patrik Schmuck, and Margarita Chli. Learning deep descriptors with scale-aware triplet networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, 2018. 1, 2, 4, 5
- [18] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014. 4, 6
- [19] Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998. 1, 7
- [20] Wen-Yan Lin, Siying Liu, Jian-Huang Lai, and Yasuyuki Matsushita. Dimensionality blessing: Clustering images by underlying distribution. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5784–5793, 2018. 2, 3
- [21] David G Lowe. Distinctive image features from scale-invariant keypoints. *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, 60(2):91–110, 2004. 2, 5, 8
- [22] Zixin Luo, Tianwei Shen, Lei Zhou, Siyu Zhu, Runze Zhang, Yao Yao, Tian Fang, and Long Quan. Geodesc: Learning local descriptors by integrating geometry constraints. In *European Conference on Computer Vision (ECCV)*, pages 170–185. Springer, 2018. 1, 2, 3, 4, 5, 6
- [23] Laurens van der Maaten and Geoffrey Hinton. Visualizing data using t-sne. *Journal of Machine Learning Research*, 9(Nov):2579–2605, 2008. 7
- [24] Krystian Mikolajczyk and Cordelia Schmid. A performance evaluation of local descriptors. *IEEE PAMI*, 27(10):1615–1630, 2005. 2
- [25] Krystian Mikolajczyk and Cordelia Schmid. A performance evaluation of local descriptors. *IEEE PAMI*, 27(10):1615–1630, 2005. 2
- [26] Anastasiia Mishchuk, Dmytro Mishkin, Filip Radenovic, and Jiri Matas. Working hard to know your neighbor’s margins: Local descriptor learning loss. In *Advances in Neural Information Processing Systems (NIPS)*, pages 4826–4837, 2017. 1, 2, 3, 4, 5
- [27] Hyun Oh Song, Yu Xiang, Stefanie Jegelka, and Silvio Savarese. Deep metric learning via lifted structured feature embedding. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4004–4012, 2016. 2
- [28] Adam Paszke, Sam Gross, Soumith Chintala, Gregory Chanan, Edward Yang, Zachary DeVito, Zeming Lin, Al-

- ban Desmaison, Luca Antiga, and Adam Lerer. Automatic differentiation in pytorch. In *NIPS-W*, 2017. 4
- [29] Filip Radenović, Giorgos Tolias, and Ondřej Chum. Cnn image retrieval learns from bow: Unsupervised fine-tuning with hard examples. In *European Conference on Computer Vision (ECCV)*, pages 3–20. Springer, 2016. 1
- [30] Torsten Sattler, Will Maddern, Akihiko Torii, Josef Sivic, Tomáš Pajdla, Marc Pollefeys, and Masatoshi Okutomi. Benchmarking 6dof urban visual localization in changing conditions. *CoRR*, abs/1707.09092, 2017. 1
- [31] Johannes L Schonberger and Jan-Michael Frahm. Structure-from-motion revisited. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4104–4113, 2016. 1, 2
- [32] Johannes Lutz Schönberger, Hans Hardmeier, Torsten Sattler, and Marc Pollefeys. Comparative evaluation of hand-crafted and learned local features. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017. 4, 6, 7
- [33] Johannes L Schönberger, Enliang Zheng, Jan-Michael Frahm, and Marc Pollefeys. Pixelwise view selection for unstructured multi-view stereo. In *European Conference on Computer Vision (ECCV)*, pages 501–518. Springer, 2016. 1
- [34] Edgar Simo-Serra, Eduard Trulls, Luis Ferraz, Iasonas Kokkinos, Pascal Fua, and Francesc Moreno-Noguer. Discriminative learning of deep convolutional feature point descriptors. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*. 1, 2, 4, 5
- [35] Karen Simonyan, Andrea Vedaldi, and Andrew Zisserman. Learning local feature descriptors using convex optimisation. *IEEE PAMI*, 36(8):1573–1585, 2014. 2
- [36] Yurun Tian, Bin Fan, Fuchao Wu, et al. L2-net: Deep learning of discriminative patch descriptor in euclidean space. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, volume 1, page 6, 2017. 1, 2, 3, 4, 5, 7
- [37] Tomasz Trzcinski, Mario Christoudias, Pascal Fua, and Vincent Lepetit. Boosting binary keypoint descriptors. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2874–2881, 2013. 2
- [38] Zhenhua Wang, Bin Fan, and Fuchao Wu. Local intensity order pattern for feature description. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, pages 603–610. IEEE, 2011. 2
- [39] Simon Winder, Gang Hua, and Matthew Brown. Picking the best daisy. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 178–185. IEEE, 2009. 2, 4
- [40] Sergey Zagoruyko and Nikos Komodakis. Learning to compare image patches via convolutional neural networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4353–4361, 2015. 2
- [41] Xu Zhang, X Yu Felix, Sanjiv Kumar, and Shih-Fu Chang. Learning spread-out local feature descriptors. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, pages 4605–4613, 2017. 2, 4, 5, 7
- [42] Xuefei Zhe, Shifeng Chen, and Hong Yan. Directional statistics-based deep metric learning for image classification and retrieval. *arXiv preprint arXiv:1802.09662*, 2018. 4
- [43] Feng Zhou and Fernando De la Torre. Factorized graph matching. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 127–134. IEEE, 2012. 1, 2