Learning Local Image Descriptors with Deep Siamese and Triplet Convolutional Networks by Minimising Global Loss Functions

Vijay Kumar B G, Gustavo Carneiro, Ian Reid
The University of Adelaide, SA 5005, Australia
{vijay.kumar,gustavo.carneiro,ian.reid}@adelaide.edu.au

Abstract

Recent innovations in training deep convolutional neural network (ConvNet) models have motivated the design of new methods to automatically learn local image descriptors. The latest deep ConvNets proposed for this task consist of a siamese network that is trained by penalising misclassification of pairs of local image patches. Current results from machine learning show that replacing this siamese by a triplet network can improve the classification accuracy in several problems, but this has yet to be demonstrated for local image descriptor learning. Moreover, current siamese and triplet networks have been trained with stochastic gradient descent that computes the gradient from individual pairs or triplets of local image patches, which can make them prone to overfitting. In this paper, we first propose the use of triplet networks for the problem of local image descriptor learning. Furthermore, we also propose the use of a global loss that minimises the overall classification error in the training set, which can improve the generalisation capability of the model. Using the UBC benchmark dataset for comparing local image descriptors, we show that the triplet network produces a more accurate embedding than the siamese network in terms of the UBC dataset errors. Moreover, we also demonstrate that a combination of the triplet and global losses produces the best embedding in the field, using this triplet network. Finally, we also show that the use of the central-surround siamese network trained with the global loss produces the best result of the field on the UBC dataset.

1. Introduction

The design of effective local image descriptors has been instrumental for the application of computer vision methods in several problems involving the matching of local image patches, such as wide baseline stereo [21], structure from motion [22], image classification [19, 29], just to name a few. Over the last decades, numerous hand-crafted [8, 19, 25] and automatically learned [3, 4, 10, 12, 20, 28, 32, 36] local image descriptors have been proposed and used in the applications above. Despite their conceptual differences, these two types of local descriptors are formed based on similar goals: descriptors extracted from local image patches of the same 3-D location of a scene must be unique (compared with descriptors from different 3-D locations) and robust to brightness and geometric deformations. Given the difficulty in guaranteeing such goals for hand-crafted local descriptors [8, 19, 25], the field has gradually focused more on the automatic learning of such local descriptors, where an objective function that achieves the goals above is used in an optimisation procedure. In particular, the most common objective function minimises the distance between the descriptors from the same 3-D location (*i.e.*, same class) extracted under varying imaging conditions and different viewpoints and, at the same time, maximises that distance between patches from different 3-D locations (or different classes) [3, 4, 10, 12, 20, 27, 28, 32, 36].

The more recently proposed approaches [10, 12, 20, 36] based on deep ConvNets [18] optimise slightly new objective functions that have the same goal as mentioned above. Specifically, Zagoruyko and Komodakis [36] and Han *et al*. [12] minimise a pairwise similarity loss of local image patches using a siamese network [2] (see Fig. 1-(b)), where the patches can belong to the same or different classes (a class is for example a specific 3-D location). Dosovitskiy *et al*. [10] minimise a multi-class classification loss (Fig. 1-(c)), where the model outputs the classification of a single input patch into one of the many descriptor classes (estimated in an unsupervised manner). Moreover, Masci *et al*. [20] propose a siamese network trained with a pairwise loss that minimises the distance (in the embedded space) between patches of the same class and maximises the distance between patches of different classes (Fig. 1-(b)). Even though these methods show substantial gains compared to the previous state of the art in public benchmark datasets [3, 4, 5, 24, 28, 31, 32], we believe that the loss functions and network structures being explored for this task can be improved. For instance, the triplet network [33, 14, 26, 35] (see Fig. 1-(d)) has been shown to improve the siamese network on several classification problems, and the training of the siamese and triplet networks can involve loss functions based on global classification results, which has the potential to generalise better.

In this paper, we propose the use of the triplet net-

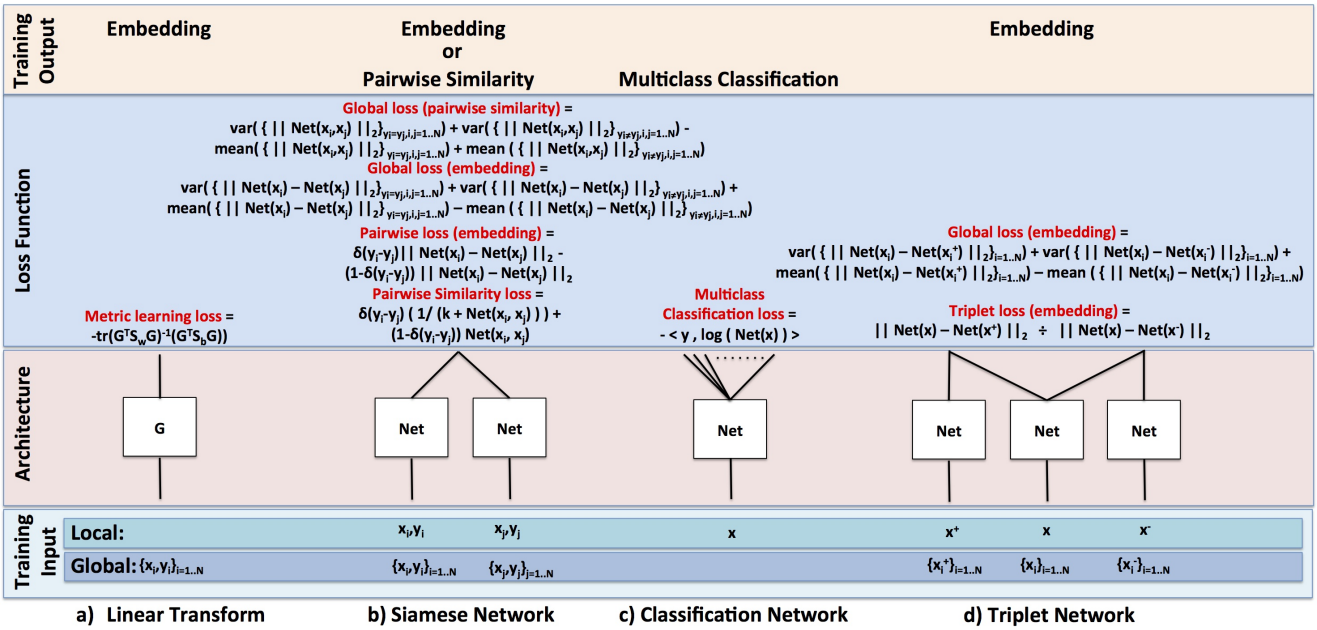


Figure 1. Comparison between different types of loss functions, network architectures and input/output types used by training methods of local image descriptor models. The metric learning loss to learn a linear transform $\mathbf{G}$ is represented in (a) [3, 4, 28, 32] (please see text for the definition of $\mathbf{S}_w$ and $\mathbf{S}_b$) and produces a feature embedding; in (b) we show the siamese network [12, 20, 36] that can be trained with different loss functions and input types, where $\delta(.)$ denotes the Dirac delta function, y is the data label, Net(x) represents the ConvNet response for input x (similarly for Net(x_i,x_j)), and the output can be an embedding (*i.e.*, Net(x)) or a pairwise similarity estimation (*i.e.*, Net(x_i,x_j)); the classification network in (c) can be used when classes of local image descriptors can be defined [10] and used in a multiclass classification problem; and in (d) the recently proposed triplet network [33, 14, 26, 35] is displayed with different loss functions and input types, where x^+ represents a point belonging to the same class as x and x^- a point from a different class of x (this triplet net produces in general an embedding). Note that our proposed global loss (embedding) in (b) and (d) takes the whole training set as input and minimises the variance of the distance of points belonging to the same and different classes and at the same time, minimise the mean distance of points belonging to the same class and maximise the mean distance of points belonging to different classes. The global loss (pairwise similarity) in (b) is similarly defined (please see text for more details).

work [33, 14, 26, 35] (Fig. 1-(d)) and a new global loss function to train local image descriptor learning models that can be applied to the siamese and triplet networks (Fig. 1-(b),(d)). The global loss to produce a feature embedding minimises the variance of the distance between descriptors (in the embedded space) belonging to the same and different classes, minimises the mean distance between descriptors belonging to the same class and maximises the mean distance between descriptors belonging to different classes (Fig. 1-(b),(d)). For the case of pairwise similarity in siamese networks, this global loss minimises the variances of the pairwise similarity between descriptors belonging to the same and different classes, maximises the mean similarity between descriptors belonging to the same class and minimises the mean similarity between descriptors belonging to different classes (Fig. 1-(b)). We first extend the siamese network [12, 20, 36] to a triplet network, trained with a triplet loss [33, 14, 26, 35] and regularised by the proposed global loss (embedding). Then we take the siamese network [12, 20, 36] and train it exclusively with the global loss (pairwise similarity). Finally, we take the central-surround siamese network [36], which is the current state-of-the-art model for the problem of local image descriptor learning, and train it with the global loss (pairwise similarity). Given that we use stochastic gradient descent (SGD) for learning the network, we approximate the global loss by computing the statistics of the mini-batch. We show on the public benchmark UBC dataset [1, 3, 30] that: 1) the triplet network shows better classification results than the siamese network [2, 20, 36]; 2) the combination of the triplet and the global loss functions improves the results produced by the triplet loss from item (1) above, resulting in the best embedding result in the field for the UBC dataset; and 3) the global loss function used to train the central-surround siamese network [36] produces the best classification result on the UBC dataset.

2. Related Work

In this section, we first discuss metric learning methods, which form the basis for several local image descriptor learning approaches. Then, we discuss relevant local image descriptor learning methods recently proposed in the field, and highlight our contributions.

2.1. Metric Learning

In general, metric learning (see Fig. 1-(a)) assumes the existence of a set of points represented by $\{\mathbf{x}_i\}_{i=1}^N$, with $\mathbf{x}_i \in \mathbb{R}^n$ and a respective set of labels $\{y_i\}_{i=1}^N$, with $y_i \in \{1, ..., C\}$, and the goal is to find a Mahalanobis distance with parameter $\mathbf{W}$. For example, the square distance between $\mathbf{x}_i$ and $\mathbf{x}_j$ is [16, 34]:

$$d_{\mathbf{W}} = (\mathbf{x_i} - \mathbf{x_j})^\top \mathbf{W} (\mathbf{x_i} - \mathbf{x_j}), \quad (1)$$

where the factorisation of the matrix $\mathbf{W} = \mathbf{G}\mathbf{G}^\top$ (with $\mathbf{G} \in \mathbb{R}^{n \times m}$) allows us to formulate the following optimisation problem: $\mathbf{G}^* = \arg\max_{\mathbf{G}} \text{tr}\left((\mathbf{G}^\top \mathbf{S}_w \mathbf{G})^{-1} (\mathbf{G}^\top \mathbf{S}_b \mathbf{G})\right)$, with $\mathbf{S}_k = \sum_{ij} \mathbf{Y}_k (\mathbf{x_i} - \mathbf{x_j})(\mathbf{x_i} - \mathbf{x_j})^\top$, $\mathbf{Y}_w = \mathbf{Y}$, $\mathbf{Y}_b = 1 - \mathbf{Y}$, and $\mathbf{Y}_{ij} = 1$ if $y_i = y_j$ and $\mathbf{Y}_{ij} = 0$, otherwise. This optimisation is solved using the generalised Eigenvalue problem, which generates a linear feature transform that effectively produces a feature embedding. The method above has been extended in many ways, such that: 1) it can handle multimodal distributions in [31]; 2) it optimises K nearest neighbour classification, which is formulated as a softmax loss minimisation and estimates a linear transform with eigenvalue decomposition [11]; 3) it optimises a large margin re-formulation of the problem in (1) using semidefinite programming [34]; 4) it can use a prior for $\mathbf{W}$, which regularises the training and gets around the cubic complexity issues of the previous methods [9]; and 5) it can be extended to large problems using equivalence constraints [17]. However, the main issue is the fact that (1) leads to a linear transformation that is unlikely to handle some of the difficult (and usually more interesting) learning problems.

Extending (1) to a non-linear transformation can be done by re-formulating $\mathbf{S}_k$ such that it involves inner products, which can then be kernelised [31], and the optimisation is again solved with generalised Eigenvalue problem [31]. Alternatively, this non-linear transform can be learned with a ConvNet using a siamese network [2] that minimises a pairwise loss [6] (Fig. 1-(b)) by reducing the distance of patches (in the embedded space) belonging to the same class and increasing the distance of patches from different classes, similarly to the objective function derived from (1). Note that this siamese network can produce either an embedding or a pairwise similarity estimation, depending on the architecture and loss function. This siamese network has been extended to a triplet network that uses a triplet loss [33, 14, 26, 35] (Fig. 1-(d)), which has been shown not only to produce the best classification results in several problems (e.g., STL10 [7], LineMOD [13], Labelled Faces in the Wild), but also to produce effective feature embeddings.

2.2. Local Image Descriptor

In the past, many local image descriptor learning methodologies have been proposed, with most based on the linear or non-linear distance metric learning, and explored in different ways [3, 4, 28, 32]. However, these methods have been shown to produce significantly worse classification results on the UBC dataset [1, 3, 30] than the recently proposed siamese deep ConvNets [12, 20, 36] (note that the UBC dataset is a benchmark dataset that has been used to compare local image descriptors). Even though the triplet network [33, 14, 26, 35] has been demonstrated to improve the results produced by the siamese networks, it has yet to be applied to the problem of local image descriptor learning. Finally, another relevant method is the discriminative unsupervised learning of local descriptors [10], which uses a single deep ConvNet to classify input local patches into many classes, which are generated in an unsupervised manner (Fig. 1-(c)). However, the latter method has not been applied to the UBC dataset mentioned above. It is also important to notice that none of the deep ConvNets methods above use the whole training set in a global loss function during the learning process, which can improve the generalisation ability of the model.

3. Methodology

As mentioned above in Sec. 2.1, we assume the existence of a training set of image patches and their respective classes, *i.e.*, $\{(\mathbf{x}_i, y_i)\}_{i=1}^N$, with $\mathbf{x}_i \in \mathbb{R}^n$ and $y_i \in \{1, ..., C\}$ (note that we use n as the patch size to simplify the notation, but the extension to a matrix representation for $\mathbf{x}$ is trivial). The first goal of our work is to use a triplet network and respective triplet loss (defined below in detail) [33, 14, 26, 35] to produce a feature embedding $f(\mathbf{x}, \theta_f)$ defined by $f : \mathbb{R}^n \times \mathbb{R}^k \rightarrow \mathbb{R}^m$, where $\theta_f \in \mathbb{R}^k$ denotes the network parameters (weights, biases and etc.). The second goal is to design a new global loss function to train the triplet [33, 14, 26, 35] and siamese networks [12, 20, 36], where we are particularly interested in the 2-channel 2-stream network, represented by a multi-resolution central-surround siamese network. Essentially, the siamese network can form a feature embedding, like the one above, or a pairwise similarity estimator, represented with $g(\mathbf{x}_i, \mathbf{x}_j, \theta_g)$, which is defined by $g : \mathbb{R}^n \times \mathbb{R}^n \times \mathbb{R}^k \rightarrow \mathbb{R}$. In this section, we first explain the siamese and triplet networks, then we describe the proposed global loss function, and we also present the models being proposed in this paper.

3.1. Siamese and Triplet Networks

The siamese network [2, 12, 20, 36] is trained with a two-tower deep ConvNet (Fig. 1-(b)), where the weights on both towers are initialised at the same values and during stochastic gradient descent, they receive the same gradients (*i.e.*, the weights on both towers are tied). We consider the following definition of a deep ConvNet:

$$f(\mathbf{x}, \theta_f) = f_{\text{out}} \circ r_L \circ h_L \circ f_L \circ ... \circ r_1 \circ h_1 \circ f_1(\mathbf{x}), \quad (2)$$

where the parameter θ_f is formed by the network weight matrices, bias vectors, and normalisation parameters for

each layer $l \in \{1, ..., L\}$, $f_l(.)$ represents the pre-activation function (*i.e.*, the linear transforms in the convolutional layers), $h_l(.)$ represents a normalisation function, and $r_l(.)$ is a non-linear activation function (e.g., ReLU [23]). Also note that $\mathbf{f}_l = [\mathbf{f}_{l,1}, ..., \mathbf{f}_{l,n_l}]$ is an array of n_l pre-activation functions, representing the number of features in layer l. A siamese network is then represented by two identical deep ConvNets trained using pairs of labelled inputs, where one possible loss function (called pairwise loss) minimises the distance between embedded features of the same class and maximises the distance between embedded features of different classes, as follows [6, 20]:

$$J_1^s(\mathbf{x}_i, \mathbf{x}_j, \theta_f) = \delta(y_i - y_j)\|f^{(1)}(\mathbf{x}_i, \theta_f) - f^{(2)}(\mathbf{x}_j, \theta_f)\|_2 - (1 - \delta(y_i - y_j))\|f^{(1)}(\mathbf{x}_i, \theta_f) - f^{(2)}(\mathbf{x}_j, \theta_f)\|_2, \quad (3)$$

where $\delta(.)$ is the Dirac delta function and $f^{(1)}(\mathbf{x}, \theta_f)$ is constrained to equal to $f^{(2)}(\mathbf{x}, \theta_f)$. Alternatively, the siamese network can be trained as a pairwise similarity estimator, with a pairwise similarity loss that can be defined as:

$$J_2^s(\mathbf{x}_i, \mathbf{x}_j, \theta_g) = \delta(y_i - y_j)(1/(\kappa + g(\mathbf{x}_i, \mathbf{x}_j, \theta_g))) + (1 - \delta(y_i - y_j))g(\mathbf{x}_i, \mathbf{x}_j, \theta_g), \quad (4)$$

where the ConvNet $g(\mathbf{x}_i, \mathbf{x}_j, \theta_g)$ returns the similarity between the descriptors $\mathbf{x}_i$ and $\mathbf{x}_j$, with κ representing a small positive constant. Note that the loss functions used by recently proposed methods [36, 12] are conceptually similar to (4), but not exactly the same, where the idea is to produce a ConvNet $g(\mathbf{x}_i, \mathbf{x}_j, \theta_g)$ that returns large similarity values when the inputs belong to the same class and small values, otherwise. It is important to emphasise that the local descriptor learning model that currently produces the smallest error on the UBC dataset (Central-surround two-stream network) consists of a siamese network, trained with a loss similar to (4), where the input patch is sampled twice at half the resolution of the input image: one sample containing the whole patch is input to the surround stream and another sample containing a sub-patch at the centre of the original patch is input to the central stream [36]. The output of these two streams are combined to obtain a similarity score.

The triplet network [33, 14, 26, 35] (Fig. 1-(d)) is an extension of the siamese network that is trained with triplets at the input (which produces an embedding) using the triplet loss function, as follows:

$$J_1^t(\mathbf{x}, \mathbf{x}^+, \mathbf{x}^-, \theta_f) = \max\left(0, 1 - \frac{\|f^{(1)}(\mathbf{x}, \theta_f) - f^{(3)}(\mathbf{x}^-, \theta_f)\|_2}{\|f^{(1)}(\mathbf{x}, \theta_f) - f^{(2)}(\mathbf{x}^+, \theta_f)\|_2 + m}\right), \quad (5)$$

where m is the margin, $\mathbf{x}^+$ and $\mathbf{x}$ belong to the same class, $\mathbf{x}^-$ and $\mathbf{x}$ are from different classes, and $f^{(1)}(.)$, $f^{(2)}(.)$ and $f^{(3)}(.)$ are constrained to be the same network. Note that the losses in (3) and (5) are apparently similar, but they have a noticeable difference, which is the fact that a triplet of similar and dissimilar inputs gives context for the optimisation process, as opposed to the pairwise loss that the siamese network minimises (same class) or maximises (different classes) as much as possible for each pair independently [14].

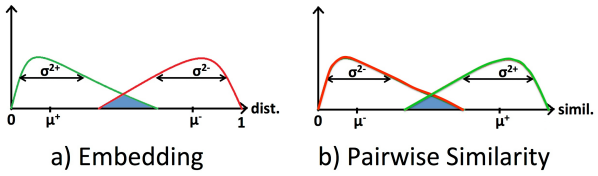


Figure 2. The objective of the proposed global loss is to reduce the proportion of false positive and false negative classification, which in the graph is represented by the area of the blue shaded region, assuming that the green curve indicates the distribution of distances in (a) or similarities in (b) between matching pairs (with mean μ^+ and variance σ^{2+}) and the red curve denotes the distribution of non-matching pair distances in (a) and similarities in (b) (with mean μ^- and variance σ^{2-}). Our proposed global loss for feature embedding (a) reduces the area mentioned above by minimising σ^{2+}, σ^{2-} and μ^+ and maximising μ^-. For the pairwise similarity in (b), the global loss minimises σ^{2+}, σ^{2-} and μ^- and maximises μ^+.

3.2. Global Loss function

The siamese and triplet networks presented in Sec. 3.1 typically contain a large number of parameters, which means that a large number of pairs or triplets must be sampled from the training data such that a robust model can be learned. However, sampling all possible pairs or triplets from the training dataset can quickly become intractable, where the majority of those samples may produce small costs in (3)-(5), resulting in slow convergence [26]. An alternative is to have a smart sampling strategy, where one must be careful to avoid focusing too much on the hard training cases because of the possibility of overfitting [26, 35, 33]. In this paper, we propose a simple, yet effective, loss function that can overcome these drawbacks.

The main idea behind our proposed loss function is the avoidance of the over- or under-sampling problems mentioned above with the assumption that the distances (or similarities) between descriptors of the same class (i.e., matching pairs) or different classes (i.e., non-matching pairs) are samples from two distinct distributions. This allows us to formulate a loss function (for the embedding case) that globally tries to: 1) minimise the variance of the two distributions and the mean value of the distances between matching pairs, and 2) maximise the mean value of the distances between non-matching pairs. Fig. 2-(a) depicts the reasoning behind the design of the proposed global loss function,

which is defined for the feature embedding case by:

$$J_1^g(\{\mathbf{x}_i\}_{i=1}^N,\{\mathbf{x}_i^+\}_{i=1}^N,\{\mathbf{x}_i^-\}_{i=1}^N,\theta_f) = (\sigma^{2+}+\sigma^{2-}) + \lambda \max\big(0,\mu^+ - \mu^- + t\big), \tag{6}$$

where $\mu^+ = \sum_{i=1}^N d_i^+/N$, $\mu^- = \sum_{i=1}^N d_i^-/N$, $\sigma^{2+} = \sum_{i=1}^N (d_i^+ - \mu^+)^2/N$, $\sigma^{2-} = \sum_{i=1}^N (d_i^- - \mu^-)^2/N$, with μ^+ and σ^{2+} denoting the mean and variance of the matching pair distance distribution, μ^- and σ^{2-} representing the mean and variance of the non-matching pair distance distribution, $d_i^+ = \frac{\|f^{(1)}(\mathbf{x}_i,\theta_f) - f^{(2)}(\mathbf{x}_i^+,\theta_f)\|_2^2}{4}$, $d_i^- = \frac{\|f^{(1)}(\mathbf{x}_i,\theta_f) - f^{(3)}(\mathbf{x}_i^-,\theta_f)\|_2^2}{4}$, λ is a term that balances the importance of each term, t is the margin between the mean of the matching and non-matching distance distributions and N is the size of the training set. Note in (6), that we assume a triplet network (i.e., $f^{(1)}(.)$, $f^{(2)}(.)$ and $f^{(3)}(.)$ are the same network), where the squared Euclidean distances of the matching and non-matching pairs of the i^{th} triplet are constrained to be $0 \leq d_i^+, d_i^- \leq 1$ because of the division by 4, and the normalisation layer enforces that the norm of the embedding is 1.

Given that we use SGD for the optimisation process, we need to derive the gradient of the global loss function, as follows:

$$\begin{aligned}
\frac{\partial J_1^g}{\partial f(\mathbf{x}_i)} &= -\frac{1}{2N}\big[2\big((d_i^+ - \mu^+)f(\mathbf{x}_i^+) + (d_i^- - \mu^-)f(\mathbf{x}_i^-)\big) \\
&\quad + \lambda(f(\mathbf{x}_i^+) - f(\mathbf{x}_i^-))\mathbb{1}((\mu^- - \mu^+) < t)\big], \\
\frac{\partial J_1^g}{\partial f(\mathbf{x}_i^+)} &= -\frac{1}{2N}\big[2\big((d_i^+ - \mu^+)f(\mathbf{x}_i)\big) \\
&\quad + f(\mathbf{x}_i)\mathbb{1}((\mu^- - \mu^+) < t)\big], \\
\frac{\partial J_1^g}{\partial f(\mathbf{x}_i^-)} &= -\frac{1}{2N}\big[2\big((d_i^- - \mu^-)f(\mathbf{x}_i)\big) \\
&\quad - f(\mathbf{x}_i)\mathbb{1}((\mu^- - \mu^+) < t)\big]
\end{aligned} \tag{7}$$

where the dependence on θ_f and the channel index $f^{(.)}$ are dropped to simplify the notation, and $\mathbb{1}(a)$ is an indicator function with value 1 when a is true. It is important to note that the gradient $\partial J_1^t/\partial f(\mathbf{x}_i)$ of the triplet loss in (5) depends only on the i^{th} triplet of the training set, whereas the gradient $\partial J_1^g/\partial f(\mathbf{x}_i)$ of the global loss in (7) depends on μ^+ and μ^-, which in turn depends on the statistics of the samples in the whole training set. This dependence on global training set statistics has the potential to suppress the spurious gradients computed from outliers and thus improving the generalisation of the trained model.

This global loss can be slightly modified to train a siamese network that estimates pairwise similarities, where the objective consists of: 1) minimising the variance of the two distributions and the mean value of the similarities between non-matching pairs, and 2) maximising the mean value of the similarities between matching pairs. Fig. 2-(b) shows the idea behind the design of the proposed pairwise similarity global loss function, which is defined by:

$$J_2^g = (\{\mathbf{x}_i\}_{i=1}^N,\{\mathbf{x}_i^+\}_{i=1}^N,\{\mathbf{x}_i^-\}_{i=1}^N,\theta_f) = (\sigma^{2+}+\sigma^{2-}) + \lambda \max\big(0, m - (\mu^+ - \mu^-)\big), \tag{8}$$

where $g(\mathbf{x},\tilde{\mathbf{x}},\theta_g)$ produces a similarity score between $\mathbf{x}$ and $\tilde{\mathbf{x}}$, $\mu^+ = \sum_{i=1}^N g(\mathbf{x}_i,\mathbf{x}_i^+,\theta_g)/N$, $\mu^- = \sum_{i=1}^N g(\mathbf{x}_i,\mathbf{x}_i^-,\theta_g)/N$, $\sigma^{2+} = \sum_{i=1}^N (g(\mathbf{x}_i,\mathbf{x}_i^+) - \mu^+)^2/N$, $\sigma^{2-} = \sum_{i=1}^N (g(\mathbf{x}_i,\mathbf{x}_i^-) - \mu^-)^2/N$, with μ^+ and σ^{2+} denoting the mean and variance of the matching pair similarity distribution, μ^- and σ^{2-} representing the mean and variance of the non-matching pair similarity distribution, λ is a term that balances the importance of each term, m is the margin between the mean of the matching and non-matching similarity distributions and N is again the size of the training set (note that we are abusing the notation with the re-definition of μ^+, μ^-, σ^{2+}, and σ^{2-}). The gradient of this global loss function is derived as

$$\begin{aligned}
\frac{\partial J_2^g}{\partial g(\mathbf{x}_i,\mathbf{x}_i^+,\theta_g)} &= \frac{2}{N}\big[(g(\mathbf{x}_i,\mathbf{x}_i^+,\theta_g) - \mu^+) \\
&\quad - \frac{1}{2}\mathbb{1}((\mu^+ - \mu^-) < m)\big] \\
\frac{\partial J_2^g}{\partial g(\mathbf{x}_i,\mathbf{x}_i^-,\theta_g)} &= \frac{2}{N}\big[(g(\mathbf{x}_i,\mathbf{x}_i^-,\theta_g) - \mu^-) \\
&\quad + \frac{1}{2}\mathbb{1}((\mu^+ - \mu^-) < m)\big].
\end{aligned} \tag{9}$$

3.3. Proposed Models

We propose **four models** for the the problem of local image descriptor learning. The **first model** consists of a triplet network trained with the triplet loss in (5), which produces an embedding - this is labelled as **TNet, TLoss**. The **second model** is a triplet network that also produces an embedding and uses the following loss function that combines the original triplet loss (5) and the proposed global loss (6) for the learning process:

$$J_1^{tg}(\{\mathbf{x}_i\}_{i=1}^N,\{\mathbf{x}_i^+\}_{i=1}^N,\{\mathbf{x}_i^-\}_{i=1}^N) = \gamma \sum_j J_1^t(\mathbf{x}_j,\mathbf{x}_j^+,\mathbf{x}_j^-) + J_1^g(\{\mathbf{x}_i\}_{i=1}^N,\{\mathbf{x}_i^+\}_{i=1}^N,\{\mathbf{x}_i^-\}_{i=1}^N), \tag{10}$$

– this model is labelled as **TNet, TGLoss**. The **third model** is a siamese network that produces a similarity estimation of an input pair of local patches, but the model is trained with the siamese global loss defined in (8) – this model is labelled as **SNet, GLoss**. Finally, the **fourth model** is the central-surround siamese network model described in Sec. 3.1 that also produces the pairwise similarity estimator of an input pair of local patches and is trained with the global loss (8) –

this model is labelled as **CS SNet, GLoss**. Note that for the first two models that produce the embedding, the comparison between local descriptors is done based on the ℓ_2 norm of the distance in the embedded feature space.

In terms of the ConvNet structure, we use an architecture similar to the one described by Zagoruyko and Komodakis [36]. Specifically, the triplet network has the following structure: B(96,7,3)-P(2,2)-B(192,5,1)-P(2,2)-B(256,3,1)-B(256,1,1)-B(256,1,1). The siamese network has the following architecture: B(96,7,3)-P(2,2)-B(192,5,1)-P(2,2)-B(256,3,1)-B(256,1,1)-C(1,1,1). Furthermore, the central-surround siamese network has the following structure: B(95,5,1)-P(2,2)-B(96,3,1)-P(2,2)-B(192,3,1)-B(192,3,1) and the final block that combines the outputs of the two input streams has components B(768,2,1)-C(1,1,1). In the description above, P(p, q) is a max pooling layer of size $p \times p$ and stride q, and B(n, k, s) is a block with the components C(n, k, s)-bnorm(n), where C(n, k, s) is a convolutional layer with n filters of kernel size k and stride s, bnorm(n) is the batch normalisation unit [15] with $2n$ parameters. Each B and C is followed by a rectified linear unit [23] except the final layer. Finally, the output feature from the exmedding networks are normalised to have unit norm, as mentioned in Sec. 3.2.

4. Toy Problem

To illustrate the robustness of the proposed global loss function to outliers, we generated a toy dataset in two dimensions with two classes (80 samples from two Gaussian distributions) represented by two distinct cloud of points, as indicated by the red and green points in Fig. 3-(a). We introduce outliers by switching the labels of randomly selected points (*i.e.*, we switch the labels of 5% of the training set, or 4 samples). We generate a set of triplets from this training set and train a ConvNet that maps the input points to an output embedding space with 128 dimensions with the following structure: B(256,2,1)-B(512,1,1)-C(128,1,1), where the output is normalised to have unit norm and these blocks are defined in Sec. 3.3. Three separate trainings are run: the first training uses the triplet loss function in (5), the second uses a combination of the triplet and global losses in (10), and the third uses only the global loss in (6). To ensure a fair comparison, we run the experiments with identical settings, where the only difference is the loss function. We evaluate the models learned from each loss function by computing the embedding of a grid of points from the input space, and labelling them based on the label of the nearest neighbour from the training set, found in the embedding space.

Figure 3-(b) shows the input space labelled according to the nearest neighbour classifier run in the embedding space generated by the triplet loss. Similarly, Fig. 3-(c) shows the same result for the combined triplet and global losses and Fig. 3-(d) displays the results for the global loss. In general, it is clear that outliers affect more the classifier in (b), which seems to be over-fitting the training data. Such labelling mistakes are reduced when we use the combination of the triplet and global losses as show in Fig. 3-(c). The label map in Fig. 3-(d) generated by the embedding that uses global loss is coherent even at the locations, where outliers can be found in the training set, indicating that the global loss function is robust to outliers.

Datasets		Proposed Models		Zagoruyko et.al.[36]		Simonyan et.al. [28]
Train	Test	TNet, TGLoss	TNet, TLoss	siam-2stream-l2	siam-l2	discr. proj.
Liberty	Notredame	**3.91**	4.47	4.54	6.01	7.22
Liberty	Yosemite	**10.65**	11.82	13.24	19.91	11.18
Notredame	Liberty	9.91	10.77	**8.79**	13.24	12.42
Notredame	Yosemite	**9.47**	10.96	13.02	12.64	10.08
Yosemite	Liberty	13.45	13.9	**12.84**	17.25	14.58
Yosemite	Notredame	**5.43**	5.85	5.58	8.38	6.82
mean		**8.8**	9.63	9.67	13.45	10.38

Table 2. **Embedding results:** False Positive Rate at 95% recall (FPR95) on UBC benchmark dataset, where bold numbers indicate the best results on the dataset. Note that for our models, we use the test set specified in [1] to compute these values.

5. Experiments

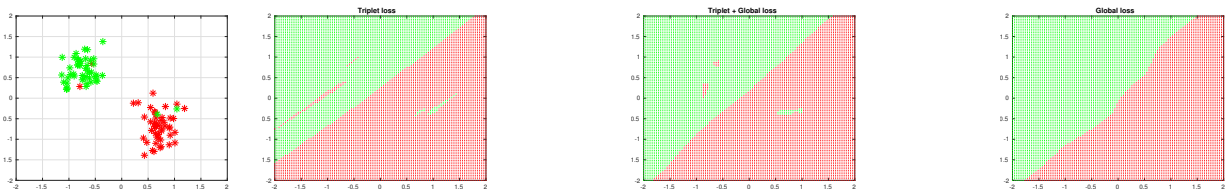
In this section, we first describe the dataset used for assessing our proposed models, then we explain the model setup, followed by a presentation of the results.

5.1. UBC Benchmark Dataset

The experiments are based on the performance evaluation of local image patches using the standard UBC benchmark dataset [1, 3, 30], which contains three sets: Yosemite, Notre Dame, and Liberty. Using these sets, we run six combinations of training and testing sets, where we use one set for training and another for testing. Each one of this sets has more than $450,000$ local image patches (with normalised orientation and scale) of size 64×64 sampled using a Difference of Gaussians (DoG) detector. In each of these sets there are more than $100,000$ patch classes that are determined based on their 3-D locations obtained from multi-view stereo depth maps. These patch classes are used to produce $500,000$ pairs of matching (*i.e.*, from the same class) and non-matching (*i.e.*, different classes) image patches. Each model is assessed using the false positive at 95% recall (FPR95) on each of the six combinations of training and testing sets, the mean over all combinations, and the receiver operating characteristic (ROC) curve also for each of the six combinations. The test set contains $100,000$ pairs with equal number of matching and non-matching pairs and is chosen as specified in [1].

5.2. Training Setup and Implementation Details

The model training is based on stochastic gradient descent (SGD) that involves: 1) the use of a learning rate



a) Training set b) Triplet loss (5) c) Combined triplet and global losses (10) d) Global loss (6)

Figure 3. Illustration to compare the robustness of different loss functions to outliers: (a) training data with outliers, (b-d) classification of points in the input space based on the nearest neighbour classifier run in the embedding space learned with the triplet loss (b), the combined triplet and global losses (c), and the global loss (d).

Datasets		Proposed Models		Zagoruyko *et al.* [36]				Xufeng *et al.* [12]	
Train	Test	CS SNet, GLoss	SNet, GLoss	2ch-2stream	2-ch	Siam	siam-2stream	4096d-F(512)	512d-F(512)
Liberty	Notredame	**0.77**	1.84	1.9	3.03	4.33	3.05	3.87	4.75
Liberty	Yosemite	**3.09**	6.61	5.00	7	14.89	9.02	10.88	13.58
Notredame	Liberty	**3.69**	6.39	4.85	6.05	8.77	6.45	6.9	8.84
Notredame	Yosemite	**2.67**	5.57	4.10	6.04	13.23	10.44	8.39	11
Yosemite	Liberty	**4.91**	8.43	7.2	8.59	13.48	11.51	10.77	13.02
Yosemite	Notredame	**1.14**	2.83	2.11	3.05	5.75	5.29	5.67	7.7
mean		**2.71**	5.28	4.19	5.63	10.07	7.63	7.75	9.82

Table 1. **Pairwise similarity results:** False Positive Rate at 95% recall (FPR95) on UBC benchmark dataset, where bold numbers indicate the best results on the dataset. Note that for our models, we use the test set specified in [1] to compute these values.

of 0.01 that gradually (automatically computed based on the number of epochs set for training) decreases after each epoch until it reaches 0.0001; 2) a momentum set at 0.9, 3) weight decay of 0.0005, and 4) data augmentation by rotating the pair of patches by 90, 180, and 270 degrees, and flipping the images horizontally and vertically (*i.e.*, augmented 5 times: 3 rotations and 2 flippings) [36]. The training set for the triplet and siamese networks consists of a set of $250,000$ triplets, which are sampled randomly from the aforementioned set of $500,000$ pairs of matching and non-matching image patches, where it is important to make sure that the triplet contains one pair of matching image patches and one patch that belongs to a different class of this pair. The mini-batch of the SGD optimisation consists of 250 triplets (randomly picked from this $250K$ set of triplets), which is used to compute the global loss in (6) and (8). Our Matlab implementation takes $\approx$ 56 hours for training a model and processes $16K$ images/sec during testing on a GTX 980 GPU.

The triplet networks **TNet-TLoss** and **TNet-TGLoss** use the three towers of ConvNets (see Fig. 1) to learn an embedding of size 256 (we choose this number of dimensions based on the feature dimensionality of the models in [36], which also have 256 dimensions before the fully connected layer). During testing, only one tower is used (all three towers are in fact the same after training) to compute the embedded features, which are compared based on the ℓ_2 norm of the distance between these embedded features. The network weights for the TNet-TLoss network are initialised randomly and trained for 100 epochs, whereas the weights for the TNet-TGLoss network are trained for 50 epochs after being initialised using the weights from TNet-TLoss network trained for 50 epochs (the initialisation from the TNet-TLoss model trained with early stopping provided a good initialisation for TNet-TGLoss). This number of epochs for training is decided based on the convergence obtained in the training set with respect to the loss function. Moreover, the margin parameter $m = 0.01$ in (5) and $\gamma = 1$, $t = 0.4$ and $\lambda = 0.8$ in (10) are estimated via cross validation. For the siamese networks **SNet-GLoss** and **CS-SNet-GLoss**, the weights are randomly initialised and trained for 80 epochs (again, based on the convergence of the training set). Finally, $m = 1$ and $\lambda = 1$ in (8) are also estimated via cross validation.

5.3. Results on UBC Benchmark Dataset

Tables 1 and 2 summarises the performance of the proposed models and compares them with the current state-of-the-art methods for the UBC dataset [1, 3, 30] using the FPR95 on each of the six combinations of training and testing sets, and the mean over all combinations. Note that we separate the results in terms of the comparison of descriptors obtained by pairwise similarity methods (Tab. 1) and embedding (Tab. 2). We also show the ROC curves for each of the six combinations of training and testing sets in Fig. 4 for our proposed models, in addition to the current state-of-the-art models [28, 36].

From the results in Tab. 2 and Fig. 4, we observe that

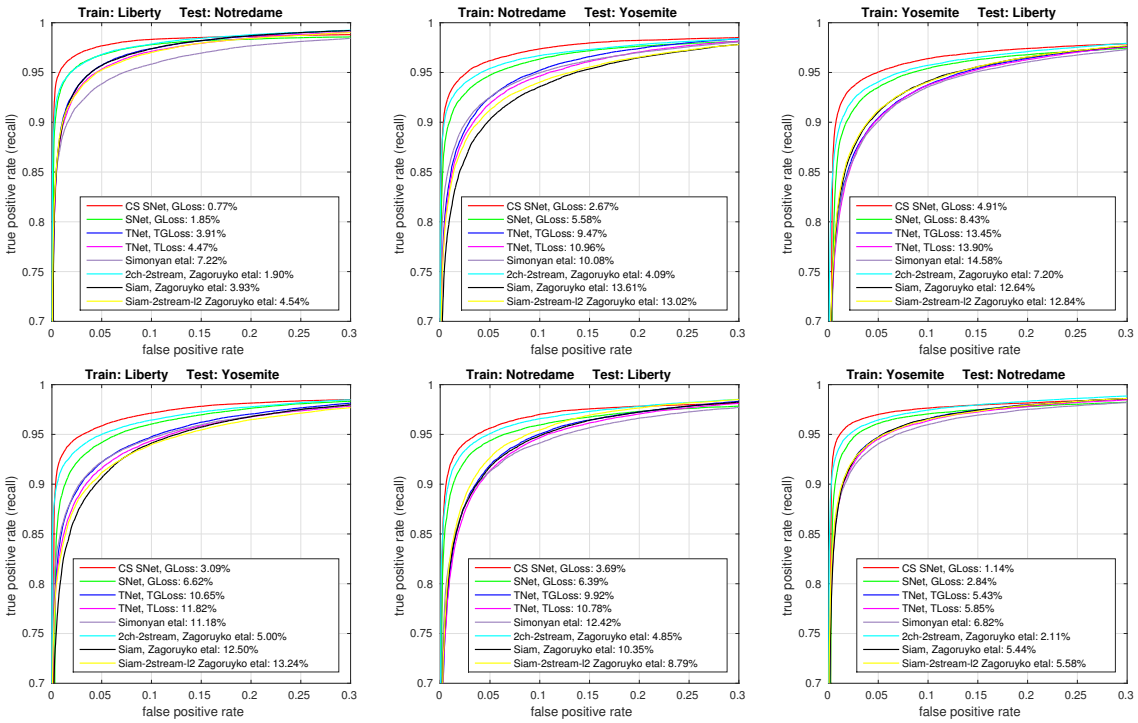


Figure 4. ROC curves on the UBC benchmark dataset for our proposed models, and the current state-of-the-art descriptors [28, 36]. Note that for our models, we use the test set specified in [1] to compute these curves, and the numbers in the legends represent the FPR95 values.

the proposed triplet network trained with a combination of the triplet and global losses (*i.e.*, the TNet-TGLoss) shows the best result in the field in terms of feature embedding. The pairwise similarity results in Tab. 1 and Fig. 4 indicate that our centre-surround siamese network trained with global loss (*i.e.*, the CS SNet, GLoss) produces a result that is almost half of the previous state-of-the-art result, *i.e.*, the 2ch-2stream [36].

Similar to [36], we notice that the siamese networks trained with the pairwise similarity loss achieve better classification performance compared to the feature embeddings produced by the triplet loss, but the dependence of the siamese networks on pairwise inputs is a potential issue during inference in terms of complexity. For instance, the ℓ_2 distance norm computation between feature embeddings can be significantly simplified to a cosine distance dot product, since the descriptor norms are equal to 1, while the siamese networks have to measure the similarity using the final fully connected (FC) layer of the network (assuming the features before that FC layer have been pre-computed). Even though pairwise similarity methods tend to perform better than feature embedding approaches, according to our results and also the results from [36], it is interesting to notice that our feature embedding model TNet-TGLoss performs better than Siam network [36] and the 512d-F(512) network [12], with both representing examples of pairwise similarity methods.

6. Conclusions

We have presented new methods for patch matching based on learning using triplet and siamese networks trained with a combination of triplet loss and global loss applied to mini-batches - this is the first time such global loss and triplet network have been applied in patch matching. This new loss overcomes a number of the issues that have previously arisen when using triplet loss, most notably slow or even unreliable convergence.

We argue that the superior results provided by our models are due to the better regularisation provided by the global loss, as shown in Sec. 4. We have shown our models to be very effective on the UBC benchmark dataset, delivering state-of-the-art results.

A natural extension of our models is the use of the global loss with the triplet network, but our preliminary results (not shown in this paper) indicate that this model does not produce better results than the ones in Table 2. We plan to extend this method to other applications, such as pre-training in visual class recognition problems.

Acknowledgements: This research was supported by the Australian Research Council through the Centre of Excellence in Robotic Vision, CE140100016, and through Laureate Fellowship FL130100102 to IDR

References

[1] Ubc patch dataset. http://www.cs.ubc.ca/~mbrown/patchdata/patchdata.html. Accessed: 2015-10-27. 2, 3, 6, 7, 8

[2] J. Bromley, J. W. Bentz, L. Bottou, I. Guyon, Y. LeCun, C. Moore, E. Säckinger, and R. Shah. Signature verification using a siamese time delay neural network. *International Journal of Pattern Recognition and Artificial Intelligence*, 7(04):669–688, 1993. 1, 2, 3

[3] M. Brown, G. Hua, and S. Winder. Discriminative learning of local image descriptors. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 33(1):43–57, 2011. 1, 2, 3, 6, 7

[4] G. Carneiro. The automatic design of feature spaces for local image descriptors using an ensemble of non-linear feature extractors. In *Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on*, pages 3509–3516. IEEE, 2010. 1, 2, 3

[5] H.-T. Chen, H.-W. Chang, and T.-L. Liu. Local discriminant embedding and its variants. In *Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on*, volume 2, pages 846–853. IEEE, 2005. 1

[6] S. Chopra, R. Hadsell, and Y. LeCun. Learning a similarity metric discriminatively, with application to face verification. In *Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on*, volume 1, pages 539–546. IEEE, 2005. 3, 4

[7] A. Coates, A. Y. Ng, and H. Lee. An analysis of single-layer networks in unsupervised feature learning. In *International conference on artificial intelligence and statistics*, pages 215–223, 2011. 3

[8] N. Dalal and B. Triggs. Histograms of oriented gradients for human detection. In *Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on*, volume 1, pages 886–893. IEEE, 2005. 1

[9] J. V. Davis, B. Kulis, P. Jain, S. Sra, and I. S. Dhillon. Information-theoretic metric learning. In *Proceedings of the 24th international conference on Machine learning*, pages 209–216. ACM, 2007. 3

[10] A. Dosovitskiy, J. T. Springenberg, M. Riedmiller, and T. Brox. Discriminative unsupervised feature learning with convolutional neural networks. In *Advances in Neural Information Processing Systems*, pages 766–774, 2014. 1, 2, 3

[11] J. Goldberger, G. E. Hinton, S. T. Roweis, and R. Salakhutdinov. Neighbourhood components analysis. In *Advances in neural information processing systems*, pages 513–520, 2004. 3

[12] X. Han, T. Leung, Y. Jia, R. Sukthankar, and A. C. Berg. Matchnet: Unifying feature and metric learning for patch-based matching. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2015. 1, 2, 3, 4, 7, 8

[13] S. Hinterstoisser, C. Cagniart, S. Ilic, P. Sturm, N. Navab, P. Fua, and V. Lepetit. Gradient response maps for real-time detection of textureless objects. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 34(5):876–888, 2012. 3

[14] E. Hoffer and N. Ailon. Deep metric learning using triplet network. *arXiv preprint arXiv:1412.6622*, 2014. 1, 2, 3, 4

[15] S. Ioffe and C. Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *Proceedings of the 32nd International Conference on Machine Learning, ICML 2015, Lille, France, 6-11 July 2015*, pages 448–456, 2015. 6

[16] P. Jain, B. Kulis, J. V. Davis, and I. S. Dhillon. Metric and kernel learning using a linear transformation. *The Journal of Machine Learning Research*, 13(1):519–547, 2012. 3

[17] M. Koestinger, M. Hirzer, P. Wohlhart, P. M. Roth, and H. Bischof. Large scale metric learning from equivalence constraints. In *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on*, pages 2288–2295. IEEE, 2012. 3

[18] A. Krizhevsky, I. Sutskever, and G. E. Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*, pages 1097–1105, 2012. 1

[19] D. G. Lowe. Object recognition from local scale-invariant features. In *Computer vision, 1999. The proceedings of the seventh IEEE international conference on*, volume 2, pages 1150–1157. Ieee, 1999. 1

[20] J. Masci, D. Migliore, M. M. Bronstein, and J. Schmidhuber. Descriptor learning for omnidirectional image matching. In *Registration and Recognition in Images and Videos*, pages 49–62. Springer, 2014. 1, 2, 3, 4

[21] J. Matas, O. Chum, M. Urban, and T. Pajdla. Robust wide-baseline stereo from maximally stable extremal regions. *Image and vision computing*, 22(10):761–767, 2004. 1

[22] N. Molton, A. J. Davison, and I. Reid. Locally planar patch features for real-time structure from motion. In *BMVC*, pages 1–10, 2004. 1

[23] V. Nair and G. E. Hinton. Rectified linear units improve restricted boltzmann machines. In *Proceedings of the 27th International Conference on Machine Learning (ICML)*, pages 807–814, 2010. 4, 6

[24] X. Niyogi. Locality preserving projections. In *Neural information processing systems*, volume 16, page 153. MIT, 2004. 1

[25] C. Schmid and R. Mohr. Local grayvalue invariants for image retrieval. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 19(5):530–534, 1997. 1

[26] F. Schroff, D. Kalenichenko, and J. Philbin. Facenet: A unified embedding for face recognition and clustering. In *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2015, Boston, MA, USA, June 7-12, 2015*, pages 815–823, 2015. 1, 2, 3, 4

[27] E. Simo-Serra, E. Trulls, L. Ferraz, I. Kokkinos, P. Fua, and F. Moreno-Noguer. Discriminative learning of deep convolutional feature point descriptors. In *The IEEE International Conference on Computer Vision (ICCV)*, December 2015. 1

[28] K. Simonyan, A. Vedaldi, and A. Zisserman. Learning local feature descriptors using convex optimisation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2014. 1, 2, 3, 6, 7, 8

[29] J. Sivic and A. Zisserman. Video google: A text retrieval approach to object matching in videos. In *Computer Vision, 2003. Proceedings. Ninth IEEE International Conference on*, pages 1470–1477. IEEE, 2003. 1

[30] N. Snavely, S. M. Seitz, and R. Szeliski. Modeling the world from internet photo collections. *International Journal of Computer Vision*, 80(2):189–210, 2008. 2, 3, 6, 7

[31] M. Sugiyama. Dimensionality reduction of multimodal labeled data by local fisher discriminant analysis. *The Journal of Machine Learning Research*, 8:1027–1061, 2007. 1, 3

[32] T. Trzcinski, M. Christoudias, P. Fua, and V. Lepetit. Boosting binary keypoint descriptors. In *Computer Vision and Pattern Recognition (CVPR), 2013 IEEE Conference on*, pages 2874–2881. IEEE, 2013. 1, 2, 3

[33] J. Wang, Y. Song, T. Leung, C. Rosenberg, J. Wang, J. Philbin, B. Chen, and Y. Wu. Learning fine-grained image similarity with deep ranking. In *2014 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2014, Columbus, OH, USA, June 23-28, 2014*, pages 1386–1393, 2014. 1, 2, 3, 4

[34] K. Q. Weinberger, J. Blitzer, and L. K. Saul. Distance metric learning for large margin nearest neighbor classification. In *Advances in neural information processing systems*, pages 1473–1480, 2005. 3

[35] P. Wohlhart and V. Lepetit. Learning descriptors for object recognition and 3d pose estimation. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015. 1, 2, 3, 4

[36] S. Zagoruyko and N. Komodakis. Learning to compare image patches via convolutional neural networks. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015. 1, 2, 3, 4, 6, 7, 8