IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received May 10, 2020, accepted July 20, 2020, date of publication August 3, 2020, date of current version August 19, 2020.

Digital Object Identifier 10.1109/ACCESS.2020.3013560

On Sensitive Minima in Margin-Based Deep Distance Learning

REZA SERAJEH[1], SEYRAN KHADEMI[2], (Member, IEEE), AMIR MOUSAVINIA[3], AND JAN C. VAN GEMERT[2]

[1]Faculty of Electrical Engineering, K. N. Toosi University of Technology, Tehran 1631714191, Iran
[2]Faculty of Electrical Engineering, Mathematics and Computer Science, Delft University of Technology, 2628 Delft, The Netherlands
[3]Faculty of Computer Engineering, K. N. Toosi University of Technology, Tehran 1631714191, Iran

Corresponding author: Amir Mousavinia (moosavie@kntu.ac.ir)

This work was supported in part by the Volkswagen Foundation through the Project ArchiMediaL, Germany, and in part by the Delft University of Technology.

ABSTRACT This paper investigates sensitive minima in popular deep distance learning techniques such as Siamese and Triplet networks. We demonstrate that standard formulations may find solutions that are sensitive to small changes and thus do not generalize well. To alleviate sensitive minima we propose a new approach to regularize margin-based deep distance learning by introducing stochasticity in the loss that encourages robust solutions. Our experimental results on HPatches show promise compared to common regularization techniques including weight decay and dropout, especially for small sample sizes.

INDEX TERMS Deep metric learning, regularization, generalization, feature point matching, contrastive loss, triplet loss.

I. INTRODUCTION

Computing meaningful distances between image pairs is difficult due to illumination, viewpoint, occlusions, etc. Current deep neural network approaches leverage training data to learn a powerful nonlinear distance measure [36]. Distance learning has many important applications such as image retrieval [10], [39], image matching [8], face verification [27], person re-identification [42], local descriptor learning [19], one-shot recognition [29], etc.

Margin-based distance learning methods using the Contrastive loss [5] or the Triplet loss [24], [32] optimize a deep network by learning a nonlinear distance where similar image pairs are optimized to be closer than dissimilar pairs. For dissimilar image pairs often a hinge-loss variant $\max(m - d, 0)$ is used so that the distance d between the dissimilar image pair is at least m away, where m is the margin. The elegance and ease of implementation makes margin-based approaches arguably the best known and popular current approach for deep distance learning [16], [19], [23], [25], [41].

In this paper we posit our observation that margin-based approaches for deep distance learning suffer from sensitive minima. As illustrated in Figure 1, there are two cases. Case 1: If the distance d of the dissimilar pair is too small, and the margin is thus violated, then gradient descent will converge to points on the margin where the loss is zero, but the gradient of d with respect to the learned parameters is high (red points), i.e.: Small changes in the parameters will lead to large changes in the distance and thus sensitivity. Case 2: The dissimilar image pair has a large enough distance d, the margin is never violated, there is no loss and thus any valid

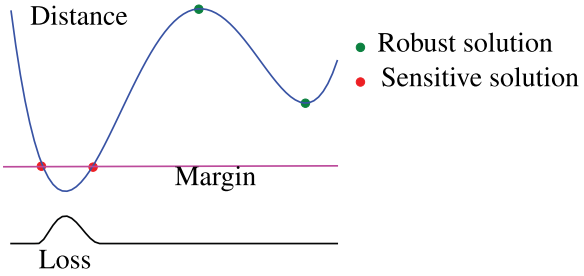
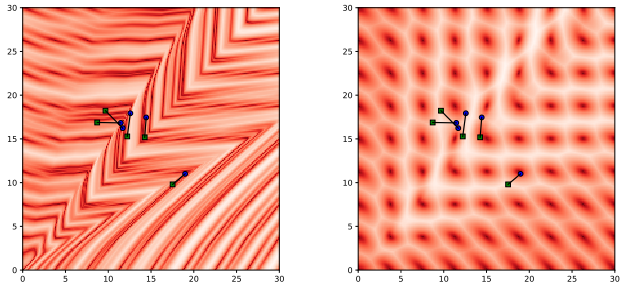


FIGURE 1. Adding a margin to distance learning creates sensitive minima. The blue line is a learned nonlinear distance function d, the purple line is the margin m, the black line is a typical distance learning hinge loss: $\max(m - d, 0)^2$. Case 1: If the margin is violated and the loss minimized with gradient decent, the solutions (red dots) have a strong gradient for d which is sensitive to small changes and thus unstable. Case 2: If the margin is not violated, any solution is satisfactory, disregarding all stability issues. In this paper we propose a method to find robust solutions (green dots).

The associate editor coordinating the review of this manuscript and approving it for publication was Tossapon Boongoen.

This work is licensed under a Creative Commons Attribution 4.0 License. For more information, see https://creativecommons.org/licenses/by/4.0/



(a) Contrastive loss (b) Stochastic Siamese (ours)

FIGURE 2. 2D example of distance learning. Five point pairs (connected by a line) are used as input to a Contrastive loss, which overfits severely to such a small sample size. In contrast, our regularized Siamese loss, trained with exactly the same network and hyper-parameters, is robust.

parameterization is selected, without taking robustness into consideration. The problem of sensitive solutions is that they do not well generalize to unseen data as changes of weights and input data are strongly correlated. Thus, using a margin for distance learning may lead to sensitive solutions which in turn causes overfitting.

We propose a novel regularization technique to combat sensitive minima for margin-based deep distance learning. We introduce stochasticity in the loss which forces the optimization to search for minima with small gradient for distance w.r.t. the parameterization, leading to robust minima. In addition, we explore the use of a robust margin not only on the dissimilar pairs, but also on similar image pairs, which prevents the network from selecting minima too close to 0, which would fit the training data too well, and likely not generalize. In Figure 2 we show a 2D example fitted on 5 random 2D pairs where the Contrastive loss severely overfits while our regularization approach is stable.

We have the following contributions. 1) We make the observation that margin-based deep distance learning suffers from sensitive minima. 2). We propose a regularization technique by introducing stochasticity in the loss that imposes non-zero gradients for sensitive minima thus preventing gradient descent to settle at those unstable solutions. 3) We show that introducing a non-zero margin for similar pairs in Siamese loss is beneficial. 4) We propose the use of square-loss instead of a hinge-loss in both Siamese and Triplet loss formulation, which leads to more robust features in the learned latent space. 5) We evaluate our regularization on recent HPatches dataset [3] and we show that compared to other popular regularization methods our regularization is particularly effective for small training size.

II. RELATED WORK

A. REGULARIZATION IN DEEP DISTANCE LEARNING

Regularized convex distance metric learning allow for generalization bounds [14], yet they do not apply to current non-convex deep distance learning. We focus on practical regularization of popular margin-based metric learning, where Siamese [5] and Triplet networks [24], [32] are best known example. Siamese networks existed in the 90s [4], but modern Siamese network with a CNN architecture dates back a decade [5]. Overfitting in Siamese networks [17], [27], [38] are treated commonly with data augmentation [19], l_2-norm normalization [28] and l_2-norm regularization [17], [38], and Dropout [42]. Triplet networks tend to outperform Siamese networks at the expense of a more complicated training process [16], [19]. There exist diverse works based on Triplet networks for feature embedding, where few examples are [1], [9], [20], [30], [40] and overfitting is addressed by regularization techniques, ranging from data augmentation [19] to more complicated sampling strategies [12], [34]. Learning local image descriptors is a well known application of deep distance learning [38]. Examples in this category are MatchNet [8], LIFT [37], HardNet [22], L2-Net [28], DOAP [10]. Typically some sort of feature normalization and batch normalization [13] are used in these networks to reduce the effect of hyper-parameters. Overfitting is reduced by dropout and limiting the architecture to few convolutional layers (only around 7 layers), without fully connected layers. All such works, including Siamese and Triplet networks typically use common regularization techniques proven in the image classification domain such as dropout, l_1/l_1-norms, etc. In contrast, we propose a regularization technique for deep metric learning specifically.

B. STOCHASTICITY FOR REGULARIZATION

Uncertainty injection is a regularization technique by making a network insensitive to small random modifications [6], which in some cases is equivalent to an analytical form, e.g., weigh decay [18] with a Gaussian prior on the input is the same as l_2-regularization. Uncertainty injection methods include dropout [26], drop-connect [31], standout [2], and shakeout [15], that are injecting Bernoulli noise to the hidden units of the deep neural network. In this context, practical Bayesian variational methods are discussed in [7]. Controversial techniques such as [35] suggest that adding noise to the output labels improves generalization. In [6, Ch. 7.5, p. 242], perturbing network weights during training is reported to approximate adding norm regularization on the gradient w.r.t. the network parameters, in the regression setting. Inspired by uncertainty injection methods we propose a regularization method by adding noise to the loss layer, yet without changing the labels.

III. METHOD

A. MARGIN-BASED DISTANCE LEARNING

Margin-based distance learning finds a mapping function from an image of size $w \times h$ with r color channels to a k-dimensional representation space, $f : I\!N^{w\times h\times r} \rightarrow I\!R^k$ where distances in $I\!R^k$ between similar image pairs d_{p} are smaller than distances to dissimilar image pairs d_{n} by a predefined margin m in the desired metric space, i.e., $d_{\mathrm{p}} + m \leq d_{\mathrm{n}}$, where

$$d_{\mathrm{p}} = d(f(I), f(I')|y = 1), \quad (1)$$

$$d_{\mathrm{n}} = d(f(I), f(I')|y = 0). \quad (2)$$

Similar image pairs have a label $y = 1$ and dissimilar image pairs the label $y = 0$. We consider a Euclidean metric space where $d(f(I), f(I'))$ outputs the Euclidean distance between two vectors of representations for an image pair $\{I, I'\}$. The mapping function f is typically parameterized by a deep convolutional neural network (CNN).

Common CNN architectures for deep distance learning are Siamese and Triplet networks. In Siamese network two identical networks, that share the same weights, are trained using a Contrastive loss function [5] which aims a zero distance for all similar image pairs and at least marginal distance m for dissimilar images:

$$J_{\text{Siam}}(\mathbf{w}, m) = \sum_{i=1}^{N_{\text{pos}}} {d_{\text{p}}^i}^2 + \sum_{j=1}^{N_{\text{neg}}} \left(\max(m - d_{\text{n}}^j, 0)\right)^2, \quad (3)$$

where N_{pos} and N_{neg} are the number of similar and dissimilar training pairs indexed by i and j. The loss is a function of the network weights w and the margin m.

A Triplet network uses three images: An anchor image is compared to a similar image and to a dissimilar image [24]. An example Triplet loss for $N = N_{\text{pos}} + N_{\text{neg}}$ images is

$$J_{\text{Trip}}(\mathbf{w}, m) = \sum_{i=1}^{N} \max\left({d_{\text{p}}^i}^2 - {d_{\text{n}}^i}^2 + m, 0\right). \quad (4)$$

B. SQUARE LOSS INSTEAD OF HINGE LOSS

Several important variants of Contrastive and Triplet losses exist in the literature [19], yet in essence they are captured by the above formulation, e.g., the hinge loss in Eqs. (3-4) can be replaced by other function of choice. The hinge loss is 0 if the margin is not violated, which makes all valid parameter instantiations equal, as illustrated in case 2 in Figure 1. To favour robust minima, we use the squared loss for both Siamese and Triplet instead of the hinge-loss.

C. NON-ZERO MARGIN FOR SIMILAR PAIRS

Our experiments show clear advantage once the desired distance for positive pairs is not set to zero unlike the original Siamese loss in Eq. (3). By allowing non-zero distances between similar images, the natural inter class variance is accounted for in the optimization process. This means although different images (patches) of positive class have same label, they are slightly different in terms of texture, illumination, viewpoint, etc. The Siamese loss of our choice is illustrated in Figure 3. In contrast to the Siamese loss, the Triplet loss does not have a fixed optimal points, i.e., m^+ (optimal point for positive pairs) and m^- (optimal point for negative pairs) alternate from sample to sample [19].

D. STOCHASTIC LOSS

Our proposed loss, referred to as Stochastic loss, is modelled by an additive random variable to the distance representation. The proposed loss for Siamese and Triplet networks are

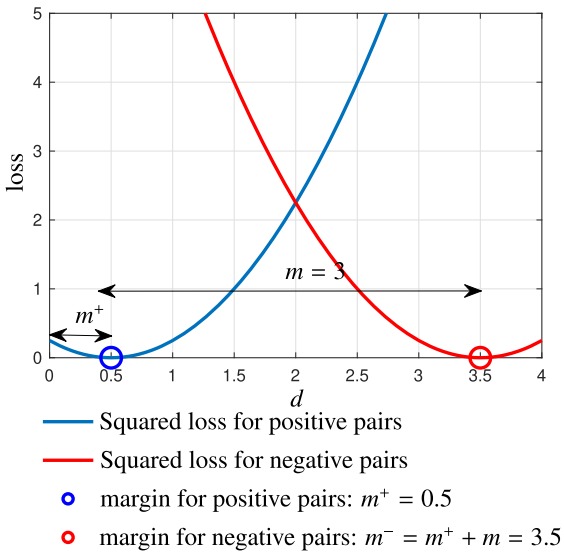


FIGURE 3. Visualization of the loss w.r.t. the distance for the squared variation of Contrastive loss in Eq. (3) with non-zero positive optimal point. The positive (negative) optimal point is depicted with the blue circle (red circle) which shows m^+ ($m^+ + m$). The relative margin is set to $m = 3$ and $m^+ = 0.5$ in this plot, therefore, the negative optimal point is set to $m^- = 3.5$.

represented as follows

$$\tilde{J}_{\text{Siam}}(\mathbf{w}, m^+, m) = \sum_{i=1}^{N_{\text{pos}}} (d_{\text{p}}^i - m^+ + \theta_p^i)^2 \quad (5)$$

$$+ \sum_{j=1}^{N_{\text{neg}}} \left(d_{\text{n}}^j - (m^+ + m) + \theta_n^i\right)^2. \quad (6)$$

$$\tilde{J}_{\text{Trip}}(\mathbf{w}, m) = \sum_{i=1}^{N} \left((d_{\text{p}}^i + \theta_p^i)^2 - (d_{\text{n}}^i + \theta_n^i)^2 - m\right)^2. \quad (7)$$

Comparing Eqs. (3-4) with Eqs. (5-7) reveals three new variables that are inserted in the proposed loss. The random variables θ_p and θ_n in Eqs. (5-7) are the core of the proposed regularization method. The constant variable m^+, in Eq. (5), is related to the non-zero optimal point (positive margin) for similar images. Note that setting the θ_p, θ_n and m^+ to zero reduces the Eqs. (5-7) to the conventional Contrastive and Triplet losses where the max-loss is replaced by squared loss.

The appended random variables θ_p and θ_n impose uncertainty to the distance of the image pair representations in Siamese and Triplet network and are considered zero mean i.i.d. random variables. This property guaranties that the injected random variables does not affect the expected value of the empirical loss. For example, $\tilde{J}_{\text{Siam}}$ will be optimized in an average sense on $\mathbb{E}\{d_{\text{p}}\} = m^+$ and $\mathbb{E}\{d_{\text{n}}\} = m^+ + m$ for positive and negative pairs, respectively. This is important because the desired optimal points for the positive and negative samples remain intact by injecting uncertainties to the loss, unlike methods that insert randomness to the hidden

layers, e.g., [6] and [26]. We keep θ_p and θ_n equal to limit the introduced hyper-parameters to one.

E. REGULARIZATION BY Stochastic LOSS

The effect of Stochastic loss on preventing the model from choosing sensitive minima is effectively explainable by exploring the behavior of the loss once optimized using gradient descent. In gradient descent, at each iteration, gradient of the loss function is computed and then, a proper step is taken towards the minimum of the loss. Taking the gradient of the Stochastic loss on positive samples in Eq. (5) leads to

$$\nabla_{\mathbf{w}}\tilde{J}^{+} = \sum_{i=1}^{N_{\text{pos}}} 2\,\nabla_{\mathbf{w}} d_{\text{p}}^{i}\left(d_{\text{p}}^{i} - m^{+} + \theta_{p}^{i}\right). \tag{8}$$

In turn, the gradient of Eq. (6) is given by

$$\nabla_{\mathbf{w}}\tilde{J}^{-} = \sum_{j=1}^{N_{\text{neg}}} 2\,\nabla_{\mathbf{w}} d_{\text{n}}^{j}\left(d_{\text{n}}^{j} - m^{-} + \theta_{n}^{j}\right), \tag{9}$$

where $\nabla_{\mathbf{w}} d_{\text{p}}^{i}$ ($\nabla_{\mathbf{w}} d_{\text{n}}^{j}$) is the gradient of the positive pair (negative pair) distance w.r.t. the network parameters, $\mathbf{w}$ for the ith (jth) training sample. Here we show that the expectation of the gradient is the same for Stochastic loss and Siamese loss. However, the difference lies in the variance that is added by stochasticity. This an important difference between norm penalty regularization techniques which change the loss expectation as well as the gradient expectation while our method affects the variance instead.

To further analyze, the sum gradient vector, expectation and covariance matrix of the gradient are

$$\frac{1}{2}\nabla_{\mathbf{w}}\tilde{J} = \sum_{i=1}^{N_{\text{pos}}} \nabla_{\mathbf{w}} d_{\text{p}}^{i}\left(d_{\text{p}}^{i} - m^{+}\right) \tag{10}$$

$$+ \sum_{j=1}^{N_{\text{neg}}} \nabla_{\mathbf{w}} d_{\text{n}}^{j}\left(d_{\text{n}}^{j} - m^{-}\right) \tag{11}$$

$$+ \sum_{i=1}^{N_{\text{pos}}} \nabla_{\mathbf{w}} d^{i}\ \theta_{p}^{i} + \sum_{j=1}^{N_{\text{neg}}} \nabla_{\mathbf{w}} d^{j}\ \theta_{n}^{j} \tag{12}$$

Eq. (12) shows the appended term to the loss gradient due to the proposed stochasticity, where Eqs. (10-11) correspond to the original Siamese loss gradient. The expectation of the gradients of proposed loss, over all training samples, is simply equal to the original Siamese loss since both θ_p and θ_n are zero-mean i.i.d. random variables.

$$\mathbb{E}\{\frac{1}{2}\nabla_{\mathbf{w}}\tilde{J}\} = \mathbb{E}\{\frac{1}{2}\nabla_{\mathbf{w}} J\}. \tag{13}$$

By Eqs. (10-13), we explicitly show the regularization as an additive term to the gradient, which reveals the similarity and difference between Siamese and Stochastic loss.

At this point, we focus on the gradient behaviour at an optimal point that is the key to understand why Stochastic loss imposes regularization on Siamese loss. The network parameters are trained when sum of the gradients are close to zero, i.e.,

$$\nabla_{\mathbf{w}}\tilde{J} = \nabla_{\mathbf{w}}\tilde{J}^{+} + \nabla_{\mathbf{w}}\tilde{J}^{-} \approx 0, \tag{14}$$

where the loss is not updated anymore in any direction. One can see, for both Siamese and Stochastic losses, the expectation of losses gradient is zero, once: 1. The scalars $\mathbb{E}\{\left(d_{\text{n}} - m^{-}\right)\}$ and $\mathbb{E}\{\left(d_{\text{p}} - m^{+}\right)\}$ (the distance from the optimal point ($\delta d_{\mathbf{w}}$)) are zero or 2. The vectors $\mathbb{E}\{\nabla_{\mathbf{w}} d_{\text{n}}\}$ and $\mathbb{E}\{\nabla_{\mathbf{w}} d_{\text{p}}\}$ (the expected metric gradient) are zero. Both options can equally make the expectation of the losses gradient zero at the optimal point. Nevertheless, only the second choice will make the variance of Stochastic loss zero, therefore we regularize the solution space by using stochaticity.

The covariance matrix of the loss gradient reveals the regularization property explicitly.

$$\Sigma[\frac{1}{2}\nabla_{\mathbf{w}}\tilde{J}] = \frac{1}{4}\mathbb{E}\{(\nabla_{\mathbf{w}}\tilde{J})(\nabla_{\mathbf{w}}\tilde{J})^{T}\} \tag{15}$$

$$= \mathbb{E}\{\theta^{2}\}\mathbb{E}\{(\nabla_{\mathbf{w}} d)(\nabla_{\mathbf{w}} d)^{T}\}, \tag{16}$$

where $\nabla_{\mathbf{w}}\tilde{J}$ is the metric gradient and $\mathbb{E}\{\theta^2\}$ is the regularization variance (we assume equal variance for θ_p and θ_n). Eq. (15) is derived assuming that variables d, $\nabla_{\mathbf{w}} d$ and θ are independent mutually. Note that $\mathbb{E}\{\theta^2\}$ is always zero in the original Siamese loss so as the covariance matrix where, this is not the case for Stochastic loss. In other words, if gradient descent converges to an optimal point meaning that there is no (or negligible) update from one iteration to another, then the Stochastic loss guaranties that the metric gradient is zero on that optimal point since the variance of the θ is designed to be non-zero. Hence, Eq. (15) admits the regularization term that promotes solutions with zero metric gradient over other (sensitive) solutions.

Note that we do not claim any convergence guarantee, however, we showed through the gradient analysis that once gradient descent converges, then the gradient of metric is always (almost) zero on that solution point. Because $\nabla_{\mathbf{w}} d_{\text{p}}$ requires to be a function of $\mathbf{w}$ so it can be tuned during the training process. Consequently, Stochastic loss does not affect a linear network that leads to a constant term for $\nabla_{w} d_{\text{p}}$. Strictly convex losses also do not benefit from Stochastic loss. Moreover, note that our proposed loss is not equivalent to adding a norm penalty of the gradient loss to the original loss as we showed that the loss expectation remains intact compared to the original loss. The same line of derivations can be shown for Triplet loss, that is omitted here for the sake of space.

IV. EXPLANATORY EXAMPLE IN 1D

For ease of explanation, we start by building intuition with an illustrative example of creating a 1D nonlinear function in Figure 4. Let the blue line be a learned nonlinear distance function for a pair of samples, for parameter values w. We wish to find the best value for w where $d(w) = m$; the black curve is the corresponding loss $(d - 1.5)^2$. The red circles show all the minima of the loss.

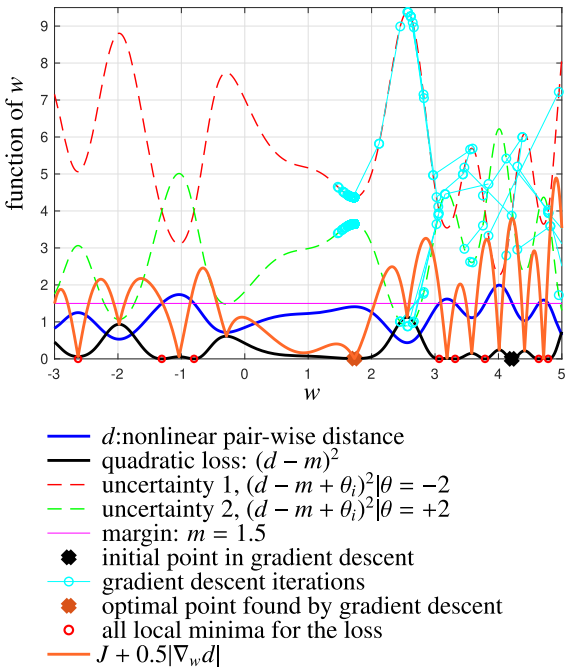


FIGURE 4. Illustration of Stochastic loss optimization for a 1D function. The blue line is a nonlinear distance function $d = f(w)$, for parameter values w. The black curve is the corresponding loss $(d-m)^2$ where $m = 1.5$. The red circles show all minima for the loss. The red and green dashed lines show the loss of adding either $-\theta$ or θ to the distance. The coloured stars indicate the random initial guess ($w_0 = 4.21$) and the minimum ($w^* = 1.74$) found by gradient descent algorithm. The orange curve shows the regularized loss with added penalty on the norm of the metric gradient. The unstable minima of the loss does not coincide with the minima of the regularized loss. Note that the minimum (or maximum) of the uncertain losses are the same as the regularized loss that are found by our method using gradient descent. The steps that are taken by the gradient descent to reach the minimum are shown by light blue circles, where, at each iteration θ is chosen randomly.

At each iteration of gradient descent, stochastically pick a constant $\in \{-\theta, \theta\}$ and simply add it to the distance d. The red and green dashed lines show these losses. In fact, the gradient descent experiences one of these two losses at each iteration randomly. Once θ is large enough compared to m then the sign of the gradients in red and green losses have opposing directions. Almost always (with the exception of $d = m$), one of the two opposed signed losses will dominate the other and pushes the gradient descent towards it's own minimum. An example of gradient descent steps for $\theta = 2$ is illustrated in Figure 4, where the initial point is randomly chosen at $w_0 = 4.21$ (black star) and the optimal point is found in 500 iterations at $w^* = 1.74$ (orange star). One can see at the sensitive optimal point, where the metric gradient is non-zero, none of the Stochastic losses have minima so there is no chance that gradient descent finds these points. In contrast the original loss can easily settle at these sensitive local minima.

We also added the conventional norm of gradient regularization to the original loss and plotted the function on Figure 4 with orange solid line. As we expect, the unstable minima

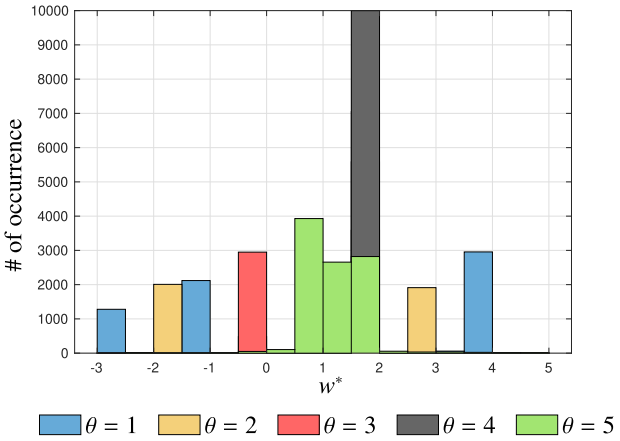


FIGURE 5. Histogram of the minima found by our method for different θ values. For $\theta = 4$, our proposed method finds the minimum at $w^* = 1.74$ with 100% chance.

in original function disappeared due to the regularization term that penalizes the loss once the norm of the gradient is non-zero. One can see that the minima of the orange function coincides with the minima of one of the Stochastic losses which confirms the regularization effect of the proposed Stochastic margin loss. In other words, the proposed approach samples the minima of the gradient regularized loss effectively without explicitly calculating the gradient of the metric. Analytical computation of the gradient is very expensive for high dimensional optimization problems as the partial derivative of a vector valued function is a matrix that needs to be calculated at every iteration of stochastic gradient descent.

In Figure 5 we perform a small experiment where gradient descent runs 10,000 times with random initialization each time for various θ values. This experiment shows the effect of hyperparameter tuning. The regularization parameter of $\theta = 4$ yields to the same (not sensitive) solution 100% of the time.

V. EXPERIMENTS

Dataset: We use HPatches [3], with more than 2.5 million image patches, which is the largest and most recent benchmark for local descriptor learning. The HPtaches benchmark is designed to evaluate image descriptors for three different tasks: patch verification, image matching and patch retrieval. The dataset is collected in various illuminations and view points from 116 different scenes with 3 level of difficulties easy, hard and tough based on the different transformation noises where 40 scenes are used for test while the other 76 scenes are considered as train data. Evaluation is done by Mean Average Precision (mAP) [3].

Architecture: Our network architecture in Table 1 consists of 11 convolutional layers (including residual blocks [11]), followed by a fully connected layer. The number of learning

TABLE 1. Our model architecture.

Layers	Filters	Output
Conv.	$7 \times 7, 32$	32×32
Res-block	$\begin{bmatrix} 6 \times 6, 32 \\ 6 \times 6, 32 \end{bmatrix}$	32×32
Conv. Avg-Pool	$6 \times 6, 64$ 2×2, stride 2	16×16
Res-block Avg-Pool	$\begin{bmatrix} 5 \times 5, 64 \\ 5 \times 5, 64 \end{bmatrix}$ 2×2, stride 2	8×8
Res-block	$\begin{bmatrix} 4 \times 4, 64 \\ 4 \times 4, 64 \end{bmatrix}$	8×8
Conv. Avg-Pool	$4 \times 4, 128$ 2×2, stride 2	4×4
Res-block	$\begin{bmatrix} 3 \times 3, 128 \\ 3 \times 3, 128 \end{bmatrix}$	4×4
FC	$128 \times 1 \times 1$	128×1

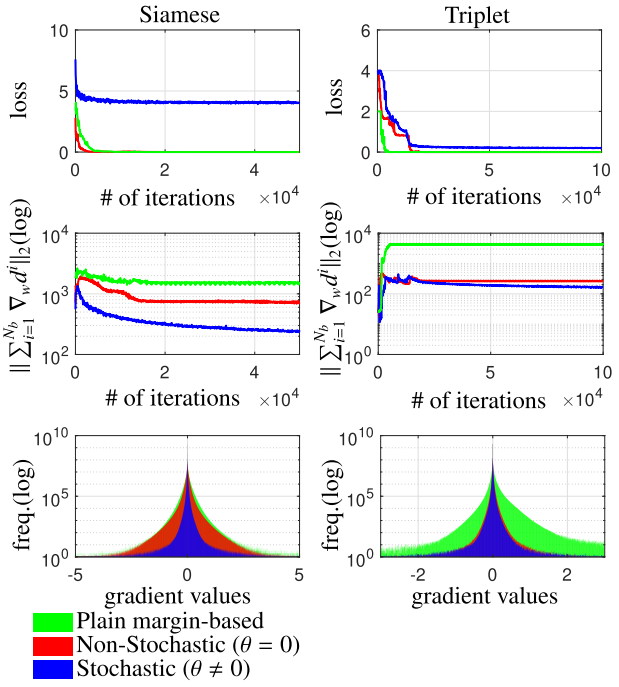


FIGURE 6. Effect of Stochastic loss on metric gradient. The plots show three states: Plain margin-based in green, Non-stochastic ($\theta = 0$) in red and Stochastic ($\theta \neq 0$) in blue for both Siamese (left) and Triplet (right). The first row is the loss over iterations and the second row is the norm of the metric gradient $||\sum_{i=1}^{N_b} \nabla_w d^i||_2$ over iterations, where N_b is number of batch pairs. The third row shows the histogram of all elements of $\nabla_w d$ for all the train pairs after training in logarithmic scale. The hyperparameter θ is set to 2 and 0.15 for Stochastic Siamese and Stochastic Triplet , respectively. Note that the metric gradient becomes smaller once stochasticity is added to the loss.

parameters for this network is approximately 1.2 M. The activation function for all the layers is Relu, padding is the "same" and average pooling is used for down sampling. Unlike other common networks, our proposed model does not use any batch normalization, dropout layer and any other conventional regularization method. Additionally, in all the experiments, we do not leverage any form of data augmentation. The network input is 32×32 gray image

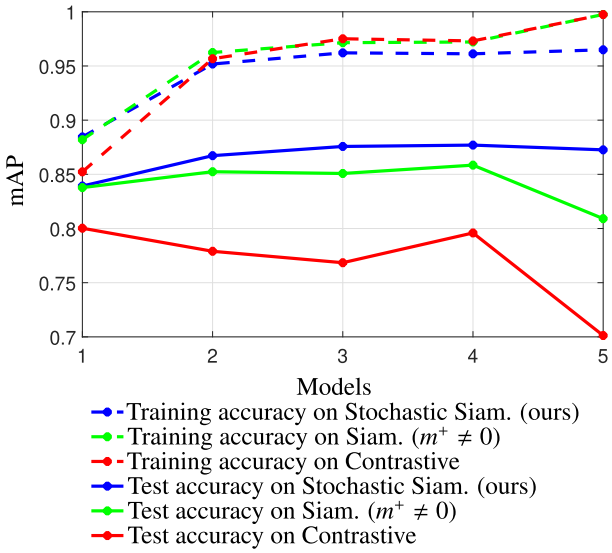


FIGURE 7. Comparison of Contrastive (red), Siamese (green), and Stochastic (blue) losses for 5 different CNN models on Hpatches dataset for patch verification task. The models 1 to 5 are respectively, correspondent to 4, 5, 8, 12 and 20 convolutional layers plus a fully connected layer at the end. Test and train mAPs, show that Stochastic loss reduces overfitting problem.

patches and the output is a 128 dimensional feature vector resembling SIFT feature descriptor [21]. For training, stochastic gradient descent with momentum of 0.9 and learning rate of 0.005 is used, where the learning rate is gradually decreased over iterations. The number of iterations in different experiments is considered enough high based on the data regime to make the loss convergence possible. The hyperparameter introduced by Stochastic loss, θ, has a zero mean Bernoulli distribution which is with $Pr(-\theta) = Pr(\theta) = 0.5$ where the θ is tuned in different experiments.

A. DOES THE METRIC GRADIENT BECOME SMALLER?

We hypothesize that for Stochastic loss shrinks the metric gradient with respect to the network weights $\nabla_w d$. We train on 3K pairs taken from Hpatches [3] dataset. Both Siamese and Triplet losses are trained in three different settings: 1) Plain margin-based, 2) non-Stochastic ($\theta = 0$) and 3) Stochastic ($\theta \neq 0$). Experiments are repeated 5 times with random initialization and averaged results are shown in Figure 6. The top row shows that networks converge, while our loss is higher because of the added stochastisity. The middle row shows that the norm of the metric gradient indeed reduces more for Stochastic loss. The histogram in the bottom row shows that the variance of $\nabla_w d$ for Stochastic loss is less than the non-Stochastic one after training for Siamese . However, for Triplet loss, the histogram of $\nabla_w d$ for Stochastic loss is closer to the non-Stochastic one, which confirms that the Triplet loss is inherently less sensitive to the weights rather than Siamese as confirmed in the literature [16], [19].

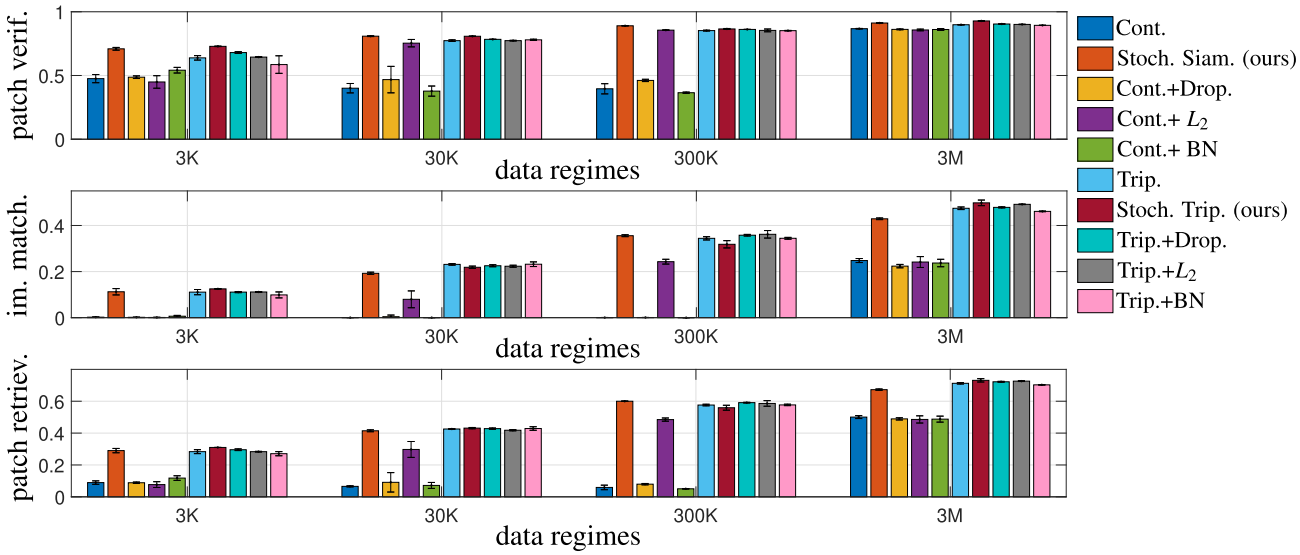


FIGURE 8. Comparison of Stochastic Siamese and Stochastic Triplet losses with conventional Contrastive and Triplet losses regularized by drop out, L_2-regularization and batch normalization (BN) on 4 different training data regimes where the *mAP* for 3 different tasks are reported. Generally, Stochastic losses result in better *mAP* especially, in the lower data regime.

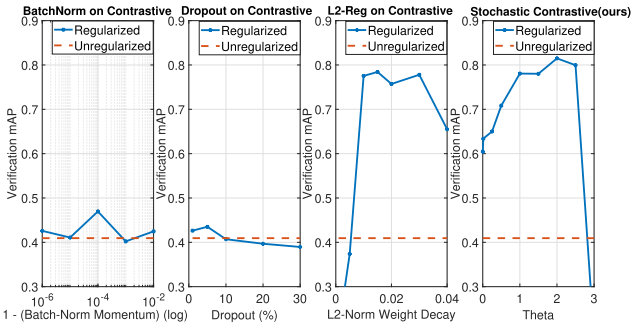


FIGURE 9. Tuning different Contrasive regularization methods.

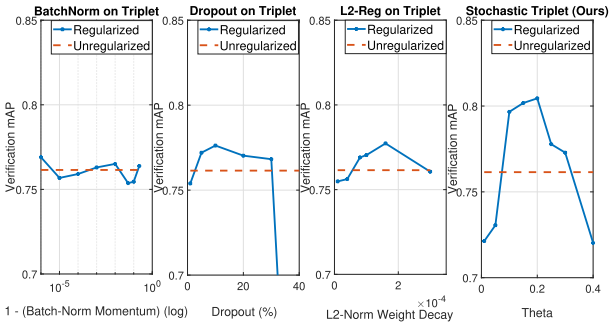


FIGURE 10. Tuning different Triplet regularization methods.

B. WHAT IS THE EFFECT ON THE ARCHITECTURE?

Since we introduce regularization, can we use deeper architectures using Stochastic loss compared to the Siamese loss? In Figure 7, The mean average precisions (mAPs) for patch verification task on Hpatches [3] dataset are shown for 5 different models. These models listed from 1 to 5 correspond to the networks with $4, 5, 8, 12$ and 20 convolutional layers plus an ultimate FC layer. We used 600K pairs collected from training portion of Hpatches dataset to learn all the models, simply without any conventional type of regularization. The stochastic parameter $\theta \in \{-1, +1\}$. Three different losses Contrastive (red), Siamese (green) and Stochastic (blue) are compared in this figure.

The first observation in Figure 7 is the significant gap between the train and test mAPs, specifically for Contrastive loss which confirms the overfitting problem, which deteriorates as the network goes deeper. The second observation shows the overall performance improvement once the squared loss is used instead of the hing loss and positive margin is added for the similar pairs (Siamese loss). In the third experiment we add the stochasticity to the Siamese loss that results in the proposed Stochastic loss. One can see that by using Stochastic loss, the accuracy gap on the train and test set is reduced even more. Additionally, it results in higher correlation between mAP of train and test data. These observations show the generalization power of Stochastic loss over different models. It is clear once the mAP on deeper model drops for Siamese and Contrastive losses, Stochastic loss retains its performance.

C. REGULARIZATION VS TRAINING SET SIZE

We conducted extensive experiments in 4 different data regimes of $3K$, $30K$, $300K$ and $3M$ from train Hpatches [3] data, on our 12-layer network. In all cases the number of positives and negatives pairs are equal. We evaluate 10 different loss settings including: Contrastive, Stochastic Siamese, Contrastive regularized by L_2-regularization, Contrastive regularized by dropout, Contrastive regularized

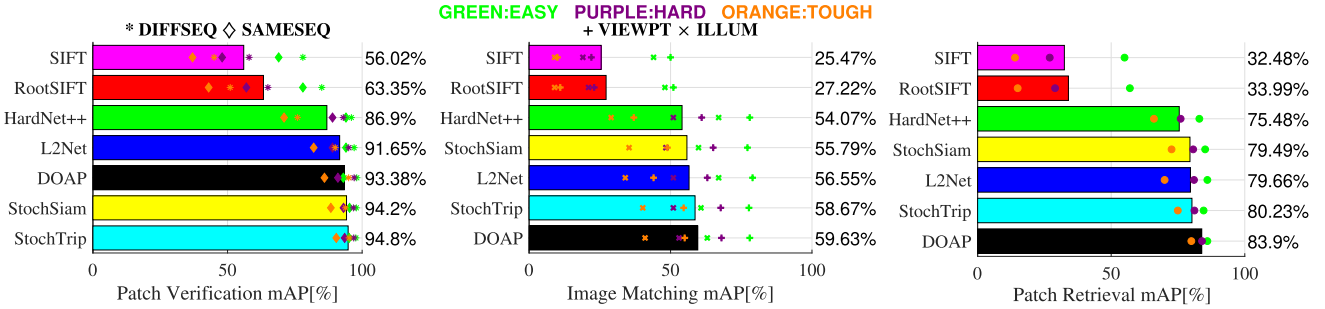


FIGURE 11. Results on Hpatches. Our proposed Stochastic losses are state of the art in patch verification task.

TABLE 2. Comparison of weights perturbation and Stochastic loss. The mAP[%]s are reported for three different Hpatches tasks.

Method	Pat. verif.	Im. match.	Pat. retriev.
Siam.	66.11	4.88	22.95
Siam.+w noise	67.24	5.64	24.17
Stoch.($\theta \neq 0$)	81.12	19.67	41.99

by batch normalization, Triplet, Stochastic Triplet, Triplet regularized by L_2-regularization, Triplet regularized by dropout and Triplet regularized by batch normalization. The evaluations are reported on three different tasks of HPatches where training process repeated 3 times with different initialization. The total number of conducted experiments for this section is $4 \times 10 \times 3 = 120$ which makes it a reliable source for comparison between different methods.

The CNN model is fixed for all the experiments and the hyperparameters are tuned on 30 K data regime using patch verification mAP. We tuned the model for all the listed methods individually, to obtain the best mAP. The result of this tuning is shown in the Figure 9 and Figure 10 where $m^+ = 1$ and $m = 2$. We report the mean and variance of each setup in the Figure 8. This experiment confirms that Stochastic Siamese loss performs better than the other conventional Contrastive loss regularized by different methods in all different tasks. However, the improvement of Triplet loss by adding stochasticity is less significant compared to the Siamese loss. One important observation from this experiment is to show the advantage of Stochastic loss when there are fewer available training data in the left part of Figure 8.

D. COMPARISON TO WEIGHT PERTURBATION

The literature [6] suggests that random Gaussian perturbation of the network weights can approximate a norm penalty regularization on the gradient, which is comparable to our proposed Stochastic loss. To compare the effect of both WP and Stochastic loss we conducted the following experiment. We train our model on three states: Siamese loss, Siamese loss with Gaussian noise is added to the weights, and Stochastic loss. After tuning the network on 30K data regime with different initializations, the best performance of these three states are reported in Table 2. The tuned Gaussian distribution of noise is $N(0, 2.5)$ and $\theta \in \{+2, -2\}$. The results show that stochasticity better generalizes for Siamese loss compared to WP.

E. COMPARISON TO OTHERS

We compare the performance of our Stochastic Siamese and Stochastic Triplet losses with other methods on HPatches [3]. We trained our model on Hpatches train dataset as same as other methods except HardNet++ which is trained on the union of Liberty [33] and HPatches. Data augmentation is not used at all. Parameters $\theta = \pm 0.75$ and $\theta = \pm 0.05$ are tuned for Stochastic Siamese and Stochastic Triplet losses, respectively. Results are shown in Figure 11. The accuracy of our Stochastic Triplet loss on patch verification task is 94.8% which introduces a new state of the art on this dataset. For image matching and patch retrieval tasks, our best mAPs are 58.67% and 80.23%, respectively which are competitive. The DOAP does better, which can be expected as the DOAP is a ranking loss which is better correlated with ranking task such as retrieval.

VI. CONCLUSION

This paper introduces new Stochastic Siamese and Stochastic Triplet losses for deep distance learning that regularizes the networks at loss layer to prevent overfitting. This is by eliminating sensitive minima, where the metric gradient is non-zero, from the loss landscape. Experimental results show the effectiveness of the proposed Stochastic loss particularly, for limited training data regime. Stochastic loss achieves state of the art on patch verification, while have a competitive performance on image matching and patch retrieval compared to the ranking losses.

REFERENCES

[1] R. Arandjelovic, P. Gronat, A. Torii, T. Pajdla, and J. Sivic, "NetVLAD: CNN architecture for weakly supervised place recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 5297–5307.

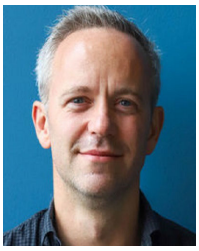
[2] L. J. Ba and B. Frey, "Adaptive dropout for training deep neural networks," in *Proc. 26th Int. Conf. Neural Inf. Process. Syst.*, vol. 2. Red Hook, NY, USA: Curran Associates Inc, 2013, pp. 3084–3092.

[3] V. Balntas, K. Lenc, A. Vedaldi, and K. Mikolajczyk, "HPatches: A benchmark and evaluation of handcrafted and learned local descriptors," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 3852–3861.
[4] J. Bromley, I. Guyon, Y. LeCun, E. Säckinger, and R. Shah, "Signature verification using a, 'siamese' time delay neural network," in *Proc. 6th Int. Conf. Neural Inf. Process. Syst.* San Francisco, CA, USA: Morgan Kaufmann Publishers Inc, 1993, pp. 737–744.
[5] S. Chopra, R. Hadsell, and Y. LeCun, "Learning a similarity metric discriminatively, with application to face verification," in *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit. (CVPR)*, vol. 1, Jun. 2005, pp. 539–546.
[6] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*. Cambridge, MA, USA: MIT Press, 2016.
[7] A. Graves, "Practical variational inference for neural networks," in *Proc. Adv. Neural Inf. Process. Syst.*, J. Shawe-Taylor, R. S. Zemel, P. L. Bartlett, F. Pereira, and K. Q. Weinberger, Eds. Red Hook, NY, USA: Curran Associates, 2011, pp. 2348–2356.
[8] X. Han, T. Leung, Y. Jia, R. Sukthankar, and A. C. Berg, "MatchNet: Unifying feature and metric learning for patch-based matching," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2015, pp. 3279–3286.
[9] B. Hariharan and R. Girshick, "Low-shot visual recognition by shrinking and hallucinating features," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Oct. 2017, pp. 3037–3046, doi: 10.1109/ICCV.2017.328.
[10] K. He, Y. Lu, and S. Sclaroff, "Local descriptors optimized for average precision," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 596–605.
[11] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 770–778.
[12] A. Hermans, L. Beyer, and B. Leibe, "In defense of the triplet loss for person re-identification," pp. 1–17, Mar. 2017, *arXiv:1703.07737*. [Online]. Available: https://arxiv.org/abs/1703.07737
[13] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," *ArXiv*, vol. abs/1502.03167, pp. 1–11, Feb. 2015.
[14] R. Jin, S. Wang, and Y. Zhou, "Regularized distance metric learning: Theory and algorithm," in *Proc. 22nd Int. Conf. Neural Inf. Process. Syst.* Red Hook, NY, USA: Curran Associates Inc, 2009, pp. 862–870, USA, 2009.
[15] G. Kang, J. Li, and D. Tao, "Shakeout: A new regularized deep neural network training scheme," in *Proc. AAAI*, 2016, pp. 1751–1757. [Online]. Available: http://www.aaai.org/ocs/index.php/AAAI/AAAI16/paper/view/11840
[16] M. Keller, Z. Chen, F. Maffra, P. Schmuck, and M. Chli, "Learning deep descriptors with scale-aware triplet networks," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 2762–2770.
[17] G. Koch, R. Zemel, and R. Salakhutdinov, "Siamese neural networks for one-shot image recognition," in *Proc. ICML Deep Learn. Workshop*, vol. 2, 2015, pp. 1–8.
[18] A. Krogh and J. A. Hertz, "A simple weight decay can improve generalization," in *Proc. Adv. Neural Inf. Process. Syst.*, J. E. Moody, S. J. Hanson, and R. P. Lippmann, Eds. San Mateo, CA, USA: Morgan Kaufmann, 1992, pp. 950–957.
[19] G. VijayKumarB., G. Carneiro, and I. D. Reid, "Learning local image descriptors with deep siamese and triplet convolutional networks by minimizing global loss functions," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Dec. 2016, pp. 5385–5394.
[20] H. Liu, J. Feng, M. Qi, J. Jiang, and S. Yan, "End-to-end comparative attention networks for person re-identification," *IEEE Trans. Image Process.*, vol. 26, no. 7, pp. 3492–3506, Jul. 2017.
[21] D. G. Lowe, "Distinctive image features from scale-invariant keypoints," *Int. J. Comput. Vis.*, vol. 60, no. 2, pp. 91–110, Nov. 2004.
[22] A. Mishchuk, D. Mishkin, F. Radenovic, and J. Matas, "Working hard to know your neighbor's margins: Local descriptor learning loss," in *Proc. NIPS*, 2017, pp. 4826–4837.
[23] Q. Qian, L. Shang, B. Sun, J. Hu, T. Tacoma, H. Li, and R. Jin, "SoftTriple loss: Deep metric learning without triplet sampling," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2019, pp. 6449–6457.
[24] F. Schroff, D. Kalenichenko, and J. Philbin, "FaceNet: A unified embedding for face recognition and clustering," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2015, pp. 815–823.
[25] H. O. Song, Y. Xiang, S. Jegelka, and S. Savarese, "Deep metric learning via lifted structured feature embedding," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 4004–4012.
[26] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Dropout: A simple way to prevent neural networks from overfitting," *J. Mach. Learn. Res.*, vol. 15, no. 1, pp. 1929–1958, 2014.
[27] Y. Taigman, M. Yang, M. Ranzato, and L. Wolf, "DeepFace: Closing the gap to human-level performance in face verification," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2014, pp. 1701–1708.
[28] Y. Tian, B. Fan, and F. Wu, "L2-Net: Deep learning of discriminative patch descriptor in Euclidean space," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 6128–6136.
[29] O. Vinyals, C. Blundell, T. Lillicrap, K. kavukcuoglu, and D. Wierstra, "Matching networks for one shot learning," in *Proc. Adv. Neural Inf. Process. Syst.*, D. D. Lee, M. Sugiyama, U. V. Luxburg, I. Guyon, and R. Garnett, Eds. Red Hook, NY, USA: Curran Associates, Inc., 2016, pp. 3630–3638.
[30] O. Vinyals, C. Blundell, T. Lillicrap, K. Kavukcuoglu, and D. Wierstra, "Matching networks for one shot learning," in *Proc. 30th Int. Conf. Neural Inf. Process. Syst.* Red Hook, NY, USA: Curran Associates Inc, 2016, pp. 3637–3645.
[31] L. Wan, M. Zeiler, S. Zhang, Y. Le Cun, and R. Fergus, "Regularization of neural networks using DropConnect," in *Proc. 30th Int. Conf. Mach. Learn.*, in Proceedings of Machine Learning Research, vol. 28, no. 3, S. Dasgupta and D. McAllester, Eds. Atlanta, GA, USA: PMLR, Jun. 2013, pp. 1058–1066. [Online]. Available: http://proceedings.mlr.press/v28/wan13.pdf
[32] 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 *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2014, pp. 1386–1393.
[33] S. Winder, G. Hua, and M. Brown, "Picking the best DAISY," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2009, pp. 178–185.
[34] C.-Y. Wu, R. Manmatha, A. J. Smola, and P. Krahenbuhl, "Sampling matters in deep embedding learning," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Oct. 2017, pp. 2859–2867.
[35] L. Xie, J. Wang, Z. Wei, M. Wang, and Q. Tian, "DisturbLabel: Regularizing CNN on the loss layer," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 4753–4762.
[36] L. Yang and R. Jin, "Distance metric learning: A comprehensive survey," Dept. Comput. Sci. Eng., Michigan State Univ., East Lansing, MI, USA, Tech. Rep., May 2006, vol. 2.
[37] K. M. Yi, E. Trulls, V. Lepetit, and P. Fua, "LIFT: Learned invariant feature transform," in *Computer Vision—ECCV*, B. Leibe, J. Matas, N. Sebe, and M. Welling, Eds. Cham, Switzerland: Springer, Sep. 2016, pp. 467–483.
[38] S. Zagoruyko and N. Komodakis, "Learning to compare image patches via convolutional neural networks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2015, pp. 4353–4361.
[39] A. Zhai and H.-Y. Wu, "Classification is a strong baseline for deep metric learning," in *Proc. BMVC*, 2019, pp. 1–12.
[40] R. Zhang, L. Lin, R. Zhang, W. Zuo, and L. Zhang, "Bit-scalable deep hashing with regularized similarity learning for image retrieval and person re-identification," *IEEE Trans. Image Process.*, vol. 24, no. 12, pp. 4766–4779, Dec. 2015.
[41] X. Zhang, F. X. Yu, S. Kumar, and S.-F. Chang, "Learning spread-out local feature descriptors," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Oct. 2017, pp. 4605–4613.
[42] Z. Zheng, L. Zheng, and Y. Yang, "A discriminatively learned CNN embedding for person reidentification," *ACM Trans. Multimedia Comput., Commun., Appl.*, vol. 14, pp. 1–20, Nov. 2018.

REZA SERAJEH received the B.Sc. degree in electronics engineering from Hakim Sabzevari University, Sabzevar, Iran, in 2010, and the M.Sc. degree in digital electronics from the Amirkabir University of Technology, Tehran, Iran, in 2013. He is currently pursuing the Ph.D. degree in electronics with the K. N. Toosi University of Technology, Tehran. He joined the Vision Laboratory, Delft University of Technology, Delft, The Netherlands, in 2017, as a Ph.D. Visiting Researcher. His research interests include computer vision, image processing, and machine learning.

SEYRAN KHADEMI (Member, IEEE) received the B.Sc. degree in electrical engineering from the University of Tabriz, in 2005, the M.Sc. degree in communications engineering from the Chalmers University of Technology, Gothenburg, Sweden, in 2010, and the Ph.D. degree from the Circuits and Systems (CAS) Group, Delft University of Technology, The Netherlands, in 2016. She was an Application Engineer with Telecommunication Company, Tehran. She was also a Postdoctoral Researcher in audio and speech processing for intelligibility enhancement with the CAS Group, Delft University of Technology, from February 2015 to 2017, where she is currently a Postdoctoral Researcher in image processing and machine learning algorithms with the Computer Vision Laboratory.

AMIR MOUSAVINIA received the B.Sc. degree (Hons.) from the Ferdowsi University of Mashhad, in 1992, the M.Sc. degree in electrical engineering from the Amirkabir University of Technology, in 1995, and the Ph.D. degree in electronics from the Iran University of Science and Technology, Tehran, Iran, in 2001. He is currently an Associate Professor with the Faculty of Computer Engineering, K. N. Toosi University of Technology, Tehran. His research interests include computer vision, multiview geometry, and digital signal processing.



JAN C. VAN GEMERT received the Ph.D. degree from the University of Amsterdam, in 2010. He was a Postdoctoral Fellow with the École Normale Supérieure, Paris. He currently Leads the Computer Vision Laboratory, Delft University of Technology, where he also teaches the M.Sc. courses in computer vision and deep learning. He has published over 75 peer-reviewed articles with more than 5000 citations. His current research interest includes adding visual inductive priors to deep learning.