Collaborative Similarity Metric Learning for Face Recognition in the Wild

Batuhan Gundogdu*, Michael J. Bianco2

1 Electrical Engineering Department, National Defense University, Naval Academy, Istanbul Turkey, 34940
2 Scripps Institution of Oceanography; University of California San Diego, La Jolla, CA, USA, 92093-0238
* E-mail: mbgundogdu@dho.edu.tr

Abstract: Utilizing different representations of face images is known to be helpful in face recognition. In this paper we propose two fusion techniques that make use of multiple face image features by collaboratively training a similarity metric learner, based on Siamese neural networks. This training procedure takes two (or possibly more) features of two face images, and outputs a similarity score that depicts whether the faces belong to the same person or not. We investigate two approaches of collaborative similarity metric learning (CoSiM), both of which are based on training Siamese neural networks jointly, as a means of early fusion. The experiments are employed on hand crafted features such as scale-invariant feature transform (SIFT) and variants of local binary pattern (LBP), on the YouTube Faces (YTF) and the Labeled Faces in Wild (LFW) datasets. We provide theoretical and empirical comparisons of the proposed models against the related methods in the literature. It is shown that the proposed technique improves on the verification accuracy, compared to single feature-based baselines. By only utilizing simple features like SIFT and LBP, the proposed techniques are shown to yield comparable results to the state-of-the-art techniques, which depend on deep convolutional architectures or higher level features.

1 Introduction

The astonishing growth in the amount of digital content volume, brought by the ever-increasing usage of the social media, has motivated the recent challenge of working on face images in the wild. In the wild images possess an increased amount of variations due to lighting, facial expression, pose, age, scale, accessories, occlusions, background and misalignment, rather than just frontal passport pictures. Such a task is generally referred to as unconstrained face recognition or verification in the literature. In this paper, we propose a collaborative (or joint) distance metric learning methodology for the task of unconstrained face verification. In face verification, the systems are required to determine if two images belong to the same person or not. The growing use of face verification technology as the new means of bio-metric security check in mobile devices increases the importance of this task.

Since the task of face verification is a pair matching problem, given a pair of face images, a measure of similarity (or dissimilarity) is evaluated to decide if they belong to the same person or not. The primary approach to this problem has concerned representation and distance metric learning (DML) or similarity metric learning (SML) methodologies to obtain a robust approach for unconstrained face images [1]. Many studies such as [2–6] investigated obtaining better representations and descriptors to facilitate face verification for given dissimilarity measures. Many other studies aimed at learning a better discriminative distance metric [7–10], while some investigated learning both distance metric and image representations together [11–13]. Most of the distance metric learning methods focus mainly on learning a linear/nonlinear function of image representations, so that a desired dissimilarity cost is minimized when two (or more) images are inputted to this function.

The most substantial work on the unconstrained face verification was initiated with the YTF [14] and LFW challenges [15]. Early work on the task focused on descriptor-based methods to conduct the same vs different discrimination using known distance metrics or classifiers such as linear discriminant analysis (LDA) or support vector machines (SVM) [16, 17]. Distance/similarity metric learning methodologies had recently been studied in the literature [18] and were brought to the unconstrained face verification task by Guillau-min et al. [7]. The primary objective in the DML methodology proposed in [7] is learning a Mahalonobis distance on the given representation space. Similarly, Nguyen and Bai [19] proposed the cosine similarity metric learning methodology in which the cosine similarity cost is optimized per kinship, in the learned projection space. Cao [20] proposed optimizing for a generalized similarity function, obtained by subtracting Euclidean distance measure from an inner product-based similarity measure calculated in two different sub-spaces to be learned. Zheng et al. [21] proposed the logistic similarity metric learning methodology which optimizes over the logistic loss function, calculated with respect to the cosine similarity and the kinship label of the images. Schroff et al. [22] proposed training deep convolutional neural networks that work directly raw images and maps them to Euclidean spaces in which the desired kinship discrimination is achieved. General application depends on using a single or a concatenation of multiple hand-crafted feature representations of images as inputs. Some studies, however, employ learning these functions from raw images using deep convolutional neural networks [13].

Distance metrics learned on a single feature representation potentially fail to utilize the complementary information brought by different features. The two main approaches to overcome this conundrum can be stated as late and early fusion techniques. In the late fusion, the fusion of information takes place in the decision level. Specifically, different similarity values are fused, each of which is obtained using a different feature representation [10]. On the other hand, in the early fusion approach, the different features are fused generally by concatenation, before the training. In this approach, several features are concatenated on the input side and a larger
In order to approach this problem, this paper offers two methodologies that would learn a similarity metric that is jointly trained, using multiple feature representations of the input images. We name the approach proposed in this paper Collaborative Similarity Metric Learning (CoSiM) since it learns a similarity metric between representations of two face images by the joint training of different features. This joint training is expected to bring a collaboration of different representations.

The literature survey provided in this section extends to Section 3, in which we provide a comparison of the methodology proposed in this paper and the related work in the literature. The detailed literature review is deferred to after Section 2, in which we introduce the methodology and the mathematical details of the proposed approach in this paper. With this set-up, we believe that the discussion of the (dis)similarities of the similar work with this paper and with each other are given in a more useful way. In Section 4, we provide information about the experiments on YTF and LFW datasets along with the experimental results. We also compare the performance of our approach with the state of the art techniques in the literature. Finally in Section 5 conclusions are drawn for this work.

1.1 Contributions of the Paper

In this paper, we propose a Siamese neural network-based SML methodology to facilitate joint learning of a similarity metric that exploits multiple feature representations of the input images. The key contributions of the paper are as follows:

- The sigma distance that was recently proposed for speech features in a keyword search task [29], is utilized for face images and compared with the other DML/SML techniques in the literature.
- Two collaborative similarity metric learning methodologies are proposed and discussed both mathematically and empirically. For experiments, two image representations: (a) scale-invariant feature transform (SIFT) and (b) local binary pattern (LBP) are combined and the SML networks are trained jointly to investigate as a means of early fusion. The proposed CoSiM networks are named mass CoSiM and attention CoSiM for the reasons to be provided along with the mathematical interpretations in Section 2.2.
- The proposed collaborative similarity metric learning methodologies are tested on two of the most challenging data sets in the literature, the YouTube Faces [14] and Labeled Faces in the Wild (LFW) [30] datasets, under the image restricted, label-free outside data and image restricted, no outside data paradigms. Results are compared with the similar work in the literature and baselines. YTF dataset is outside data restricted, label-free outside data and image sets in the literature, the YouTube Faces [14] and Labeled methodologies are tested on two of the most challenging data sets in the literature. Finally in Section 5 conclusions are drawn in Section 5.

2 Methodology

In this section, we provide mathematical and theoretical background of the model proposed.

2.1 Assumptions and Preliminaries

This paper presents an early fusion methodology for the task referred to as Similarity Metric Learning-SML.

- The "Similarity" in SML is function $f(x,y) : \mathbb{R}^d \times \mathbb{R}^d \rightarrow \mathbb{R}$ of the representations of two images $x$ and $y \in \mathbb{R}^d$, such that this function outputs a high value for images that belong to the same person, and a low value for images that belong to different people. For notational simplicity, we denote images by their vector representations $(x,y)$ and we call $x$ and $y$ friends if they are extracted from the images of the same person, and foes otherwise.

- The "Metric" in SML, is in fact a misnomer. The metric spaces are defined with a distance measure $d(x,y)$ that satisfies the axioms: (i) non-negativity, (ii) symmetry, (iii) triangle inequality and (iv) identity of discernibles [31]. On the other hand, since there is no formal definition of a "similarity metric", axioms for $f(x,y)$ are defined per application. Hence, we do not claim that we conform to the axioms of metric spaces when we call the name "similarity metric".

- The "Learning" in SML is the essence of this work. SML is actually an optimization task. Given a set of pairs $X = \{(x_i, y_i)\}$ and their labels $r_i$ denoting if they are friends or foes, for $t = 1 \cdots T$; the goal is to find a function $f(x, y)$ such that it is higher for friends than foes for all pairs that are not seen in $X$. Therefore, a proper generalization and learning methodology is desired on top of the cost minimization procedures.

As stated in the introduction section, we propose a methodology for jointly training a combination of several Siamese neural networks, which have shared layers along each feature path. For the sake of simplicity, we will be building our methodology using two of the most common image representations, i.e. (i) SIFT [32, 33] and (ii) LBP [34, 35]. However, the methodology can simply be generalized to using pairs of other features and using multiple features.

The paper is organized as follows: In Section 1 we provide an introduction and the literature review of the subject. In Section 2, the methodology is presented as well as the theoretical background. Once the notation and the objectives are set in this section, we discuss in detail the differences and similarities of this work to the literature in Section 3. In Section 4, the experiments are presented after which the conclusions are drawn in Section 5.
2.2 From Similarity Metric Learning to CoSiM

The CoSiM approach, follows the Siamese neural network structure called the sigma distance, which has recently been proposed for speech features to be used in dynamic time warping [36]. Given \( x, y \in \mathbb{R}^d \), the similarity measure for this pair of feature vectors is obtained by projection onto a new space and calculating the inner product (4). This Mahalonobis inner product is then applied to a sigmoid function so that the similarity function represents a measure of probability of the input vectors being friends [29]:

\[
f(x, y) = \sigma(x^T W^T W y + b)
\]  
(1)

where

\[
\sigma(z) = \frac{1}{1 + e^{-z}}.
\]  
(2)

\( \sigma(z) \) is the sigmoid function that reaches 1 and 0 asymptotically as \( z \) reaches \( +\infty \) and \( -\infty \), respectively. This basis SML methodology, which works for a single feature can be seen graphically in Figure 1. In training, we use the training set of triplets \((x_i, y_i, r_i)\) where \( r_i \) is the label indicating the friend-ship of the inputs \( x_i \) and \( y_i \). As for the labels \( r_i \), we use 1 for friends and 0 for foes and minimize the cross-entropy (CE) objective function. If we call the total set of parameters in the network \( \Theta \), the CE cost function is defined as:

\[
J_{CE}(\Theta; x_i, y_i, r_i) = -r_i \log(f) - (1 - r_i) \log(1 - f)
\]  
(3)

where \( f(x_i, y_i) \) is expressed as \( f \) for the sake of simplicity.

The CoSiM model, on the other hand, takes two (or more) feature representations of the images as input and outputs a similarity value, obtained by the mass CoSiM network can be expressed thus:

\[
f(x_1, y_1, x_2, y_2) = \sigma(x_1^T W_1^T W_1 y_1 + x_2^T W_2^T W_2 y_2 + b)
\]  
(4)

Using the CE cost function, we provide the gradient calculation derivations here for both of the CoSiM methodologies in order to facilitate the mathematical interpretations and intuitions that are given in the following sections:

\[
\nabla_b J = \frac{dJ}{df} \frac{df}{dz} \frac{dz}{db} = \frac{r - f}{f(1 - f)} \cdot f \cdot (1 - f) = r - f
\]  
(5)

\[
\nabla_{W_{1,2}} J = \frac{dJ}{df} \frac{df}{dz} \frac{dz}{dW_{1,2}} = (r - f) W_{1,2}(x_{1,2}^T y_{1,2} + y_{1,2} x_{1,2}^T)
\]  
(6)

where

\[
z = x_1^T W_1 y_1 + x_2^T W_2 y_2 + b
\]  
(7)

2.3 Mass CoSiM

The first topology we refer to as the mass CoSiM merges the two input after the inner product layer. There are two Siamese neural networks that are shared across the same feature representation. After both input features are projected onto their corresponding subspaces, the inner products are added before the sigmoid function application. The sigma distance depends on the similarity value obtained by the weighted inner product. These similarity values are added together in mass CoSiM, just like the accumulation of mass. The mass CoSiM architecture can be seen in Figure 2. The similarity value, obtained by the mass CoSiM network can be

2.4 Mathematical Interpretation of mass CoSiM

It is observed that the projection matrix is updated to the direction decided by the corresponding input feature (6). In other words, \( W_1 \) is updated to the direction \( W_1(x_i y_i^T + y_i x_i^T) \) and vice versa. The main advantage we obtain in adding before the sigmoid is seen on the size of this step. The update rule of the network is very similar to the single feature SML network, except with one difference. The step size of the update for each shared parameter is decided by the term \( (r - f) \), which carries the collective decision information using both features. Also, the bias term \( b \) is updated by the joint decision of the two input pairs, hence it is less susceptible to outliers and noisy decisions.

2.5 Attention CoSiM

The second topology conducts the merging of the different features, after the individual decisions of the two Siamese neural networks are made. We refer to this system as the attention CoSiM since the decisions themselves decide how much "attention" each network will receive in training. The two Siamese neural networks are still shared across the same
The mathematics of the model and the training is as follows:

\[ g(x_1, y_1, x_2, y_2) = a_1 f_1(x_1, y_1) + a_2 f_2(x_2, y_2) \]  
(8)

where \( a_1 + a_2 = 1 \). \( a_i \) are the weight parameters, which can be taken as uniform or learned through cross-validation. In this paper, we took \( a_1 = a_2 = 0.5 \)

\[ f_i(x_i, y_i) = \sigma(x_i^T W_i^T W_i y_i + b_i) \]  
(9)

The gradients with respect to the CE cost function is calculated as follows:

\[ \nabla_b_i J = \frac{dJ}{dy_i} \frac{dy_i}{dz_i} \frac{dz_i}{dW_i} \]  
\[ \nabla W_i J = \frac{dJ}{dy_i} \frac{dy_i}{dz_i} \frac{dz_i}{dW_i} \]  
(10)

where,

\[ z_i = x_i^T W_i^T W_i y_i + b_i \]  
(11)

Equation (10) follows the fact that the cross-derivatives are zero, that is

\[ \frac{df_i}{dz_j} = 0, \quad \frac{dz_i}{dW_j} = 0 \quad \text{and} \quad \frac{dz_i}{db_j} = 0 \quad \text{for} \quad j \neq i \]

By taking each partial derivative, we get

\[ \nabla b_i J = \frac{r-g}{g(1-g)} \cdot a_i \cdot f_i \cdot (1 - f_i) \]  
(12)

\[ \nabla W_i = \frac{r-g}{g(1-g)} \cdot a_i \cdot f_i \cdot (1 - f_i) W_i (x_i y_i^T + y_i x_i^T) \]  
(13)

2.6 Mathematical Interpretation of attention CoSiM

The term \((r-g)\) that appear in equations (12) and (13) also exists in bare SML [36] and mass-CoSiM in equations (5) and (6). It basically decides the size (and the direction) of the update step. In other words, if the two input vectors are friends but the system output is close to zero, the bias is increased (as much as the difference), and decreased otherwise. For the shared matrices, the projected outer products (6) are added or subtracted by this scale.

In attention CoSiM, however, the value \((r-g)\) is scaled with the following term:

\[ \eta(f_1, f_2) = a_i \frac{f_i(1 - f_i)}{g(1 - g)} \]  
(14)

\[ = a_i \frac{f_i(1 - f_i)}{(a_1 f_1 + a_2 f_2)(1 - a_1 f_1 - a_2 f_2)} \]

It is not straightforward to see how \( \eta \) contributes to the update. Apart from the pre-decided or learned attention \( a_i \), the fractional term, \( \eta \) governs the update for each channel in the Siamese neural network, based on their decisions.

When \( f_1 = f_2 \), in other words the two Siamese networks are in perfect accordance, \( \eta = 1 \) for all values. For off-diagonal values where \( f_1 \neq f_2 \), the coefficient \( \eta \) reinforces or annihilates the step according to the balance between the two decisions. Given that \( 0 \leq f_i \leq 1 \) due to the sigmoid layer, we can observe the magnitude of this term for various values of \( f_1 \) and \( f_2 \) for a better interpretation. Taking \( a_1 = a_2 = 0.5 \), the surface plot of \( \eta(f_1, f_2) \) can be seen in Figure 4.

Particularly, when \( f_1 < f_2 \) and both of the values are close to zero, it means that the one feature (say \( f_2 \)) is potentially detecting correctly and it has a large coefficient. Likewise, when \( f_1 > f_2 \) and both of the values are close to one, the coefficient is again larger than \( 1 \) (yellow edges on the surface curve). On the other hand, when there is a discrepancy between \( f_1 \) and \( f_2 \) scores, the coefficient \( \eta \) reduces as much as the discrepancy. The blue edges that touch the zero level depict such regions.

This feature of the automatic scaling of the Siamese networks brought by attention CoSiM is the reason for its name. Due to its particularly automated choice of update steps, attention CoSiM possesses a slow and distinct learning curve. The comparison of the learning curves along with the training details will be provided in the next section.
2.7 Training Topologies and Implementation Details

In our implementation, the two inputs (SIFT and LBP) are whitened by Principal Component Analysis (PCA) and dimension reduction is applied to 300 dimensions for both features. This preprocessed data was obtained from [10]. In each of the two CoSiM networks, we added a batch normalization before the shared layers for faster convergence. Batch normalization forces the input to have zero mean and unity variance.

Furthermore, a dropout of rate 0.7 was added between the batch normalization and shared layer, which was found to be crucial to avoid over-fitting to the training set. Dropout randomly zeroes a portion of the weights in training. In implementation, we used Keras toolkit with Tensorflow backend [37]. In the optimization procedure we used the adam optimizer [38] with the following parameters:

- $\alpha = 0.001$ (learning rate)
- $\beta_1 = 0.9$ ($1^{st}$ moment estimate exponential decay)
- $\beta_2 = 0.999$ ($2^{nd}$ moment estimate exponential decay)
- $\epsilon = 10^{-8}$ (denominator de-nullifier)
- $\lambda = 0.001$ (learning rate decay)

The scalar weights were initialized with random Gaussian distributed numbers. For the shared matrices, the experimental analyses showed that better convergence points were distributed numbers. For the shared matrices, the experimental analyses showed that better convergence points were distributed numbers. For the shared matrices, the experimental analyses showed that better convergence points were distributed numbers. For the shared matrices, the experimental analyses showed that better convergence points were distributed numbers.

Table 1 Mass CoSiM Neural Network Topology

<table>
<thead>
<tr>
<th>Layer</th>
<th>Input Shape</th>
<th>Output Shape</th>
<th>No.of Params</th>
<th>Connected to</th>
</tr>
</thead>
<tbody>
<tr>
<td>input, $A$,</td>
<td>300</td>
<td>-</td>
<td>0</td>
<td>-</td>
</tr>
<tr>
<td>input, $B$,</td>
<td>300</td>
<td>-</td>
<td>0</td>
<td>-</td>
</tr>
<tr>
<td>input, $J$,</td>
<td>300</td>
<td>-</td>
<td>0</td>
<td>-</td>
</tr>
<tr>
<td>batchnorm1</td>
<td>300x2</td>
<td>300x2</td>
<td>1200 ($\mu, \sigma$)</td>
<td>$A$, $B$, $J$, $B$, $J$, $J$</td>
</tr>
<tr>
<td>batchnorm2</td>
<td>300x2</td>
<td>300x2</td>
<td>1200 ($\mu, \sigma$)</td>
<td>$A$, $B$, $J$, $B$, $J$, $J$</td>
</tr>
<tr>
<td>dropout1</td>
<td>300x2</td>
<td>300x2</td>
<td>0</td>
<td>batchnorm1</td>
</tr>
<tr>
<td>dropout2</td>
<td>300x2</td>
<td>300x2</td>
<td>0</td>
<td>batchnorm2</td>
</tr>
<tr>
<td>shared1</td>
<td>300x2</td>
<td>300x2</td>
<td>9000 ($W_1$)</td>
<td>dropout1</td>
</tr>
<tr>
<td>shared2</td>
<td>300x2</td>
<td>300x2</td>
<td>9000 ($W_2$)</td>
<td>dropout2</td>
</tr>
<tr>
<td>dot1</td>
<td>300x2</td>
<td>1</td>
<td>0</td>
<td>shared1</td>
</tr>
<tr>
<td>dot2</td>
<td>300x2</td>
<td>1</td>
<td>0</td>
<td>shared2</td>
</tr>
<tr>
<td>mass</td>
<td>2</td>
<td>1</td>
<td>1 (b)</td>
<td>dot1&amp;dot2</td>
</tr>
<tr>
<td>sigmoid</td>
<td>1</td>
<td>1</td>
<td>1 (b)</td>
<td>mass</td>
</tr>
</tbody>
</table>

*note that these weights can also be learned although we took them constant due to the addition of smart attention allocation, which is also learned by the each path in the Siamese neural network.

Table 2 Attention CoSiM Neural Network Topology

<table>
<thead>
<tr>
<th>Layer</th>
<th>Input Shape</th>
<th>Output Shape</th>
<th>No.of Params</th>
<th>Connected to</th>
</tr>
</thead>
<tbody>
<tr>
<td>input, $A$,</td>
<td>300</td>
<td>-</td>
<td>0</td>
<td>-</td>
</tr>
<tr>
<td>input, $B$,</td>
<td>300</td>
<td>-</td>
<td>0</td>
<td>-</td>
</tr>
<tr>
<td>input, $J$,</td>
<td>300</td>
<td>-</td>
<td>0</td>
<td>-</td>
</tr>
<tr>
<td>batchnorm1</td>
<td>300x2</td>
<td>300x2</td>
<td>1200 ($\mu, \sigma$)</td>
<td>$A$, $B$, $J$, $B$, $J$, $J$</td>
</tr>
<tr>
<td>batchnorm2</td>
<td>300x2</td>
<td>300x2</td>
<td>1200 ($\mu, \sigma$)</td>
<td>$A$, $B$, $J$, $B$, $J$, $J$</td>
</tr>
<tr>
<td>dropout1</td>
<td>300x2</td>
<td>300x2</td>
<td>0</td>
<td>batchnorm1</td>
</tr>
<tr>
<td>dropout2</td>
<td>300x2</td>
<td>300x2</td>
<td>0</td>
<td>batchnorm2</td>
</tr>
<tr>
<td>shared1</td>
<td>300x2</td>
<td>300x2</td>
<td>9000 ($W_1$)</td>
<td>dropout1</td>
</tr>
<tr>
<td>shared2</td>
<td>300x2</td>
<td>300x2</td>
<td>9000 ($W_2$)</td>
<td>dropout2</td>
</tr>
<tr>
<td>dot1</td>
<td>300x2</td>
<td>1</td>
<td>0</td>
<td>shared1</td>
</tr>
<tr>
<td>dot2</td>
<td>300x2</td>
<td>1</td>
<td>0</td>
<td>shared2</td>
</tr>
<tr>
<td>mass</td>
<td>2</td>
<td>1</td>
<td>1 (b)</td>
<td>dot1&amp;dot2</td>
</tr>
<tr>
<td>sigmoid</td>
<td>1</td>
<td>1</td>
<td>1 (b)</td>
<td>mass</td>
</tr>
</tbody>
</table>

3 Comparison with the Similar Work in the Literature

This section mathematically compares the work with the literature and points out the similarities and dissimilarities. The notations in the original papers are altered to match the notation of this paper to facilitate direct comparison. The derivations and the formulations are coined by the referenced work, we only alter their notation to match our methodology chapter for the sake of the ease of the reader. The comparison of performances with the following works will be provided in the next section along with the experimental results.

Guillaumin et al. [7] proposed learning a Mahalonobis metric over the representation space. With the learned Mahalonobis distance (15), they define a probability that the two images belong to the same person to be as shown in (16), which is very similar to (4) in our work.

$$d_W(x, y) = (x - y)^TW(x - y)$$  \hspace{1cm} (15)

$$p = \sigma(b - d_W(x, y))$$  \hspace{1cm} (16)

This value is then used with the cross entropy cost (3), as in our work. Some of the key differences are that we learn a
similarity metric based on projected inner products and use it in the sigmoid function. That way, what corresponds to the $W$ matrix in our work is forced to be symmetric and positive definite.

Nair and Hinton [8] proposed using restricted Boltzmann machines (RBM) with rectified linear units to extract features that are used in distance metric learning. They propose calculating the cosine distance of the extracted features from the two images, which is defined in (17). They also describe the probability of belonging to the same person to be $p = \sigma(-(ud + b))$ with $u$ and $b$ being trainable parameters, which is very similar to [7] and this work.

$$d_{cos}(x, y) = 1 - \frac{x^T y}{||x|| ||y||}$$  \hspace{1cm} (17)

Nguyen and Bai [19] proposed finding a projection space on which the cosine similarity is optimized. The cosine similarity on the new space with the projection matrix $W$, is defined in to be

$$CS(x, y, W) = 1 - d_{cos}(Wx, Wy)$$  \hspace{1cm} (18)

The shared projection matrix is very similar to the methodology in this work. While in [19] the target-output control is done by the L2-normalization of the cosine distance, this work uses the sigmoid non-linearity. Furthermore, Nguyen and Bai use a hinge-like loss to obtain an optimized projection matrix, whereas we use cross entropy

$$J(W) = \sum_{t \in friends} CS(x_t, y_t, W) - \alpha \sum_{t \in foes} CS(x_t, y_t, W) - \beta ||W - W_o||^2_2$$  \hspace{1cm} (19)

The last term serves for regularization and it is interesting to note that they used an identity matrix for $W_0$, which turned out to be the empirically obtained best initializer of this work.

In another similarity metric learning-based work, Cao et al. proposed a convex generalized similarity function, to address the non-convex behavior of the L2-normalized cosine distance, described in (17) [10, 20]. Using the learned Mahalanobis distance in (15), a projected inner product similarity $s_W(x, y)$ is defined as follows:

$$s_W(x, y) = x^T W y$$  \hspace{1cm} (20)

The convex generalized similarity metric is then defined as :$$f_{\{w_1, w_2\}}(x, y) = s_W(x, y) - d_{w_2}(x, y)$$  \hspace{1cm} (21)

Using this generalized similarity measure, the learning problem is done via the following constraint optimization procedure

$$\min_{w_1, w_2} \sum_{x_t, y_t \in X} \xi_t + \frac{\gamma}{2} (||W_1 - I||^2_2 + ||W_2 - I||^2_2), \hspace{1cm} s.t. \hspace{1cm} r_{f_{\{w_1, w_2\}}}(x_t, y_t) \geq 1 - \xi_t, \hspace{1cm} \xi_t \geq 0$$  \hspace{1cm} (22)

where $\xi_t$’s are the slack variables. One other difference from our work is that they take $r_t \in \{-1, 1\}$, while we take $r_t \in \{0, 1\}$ as the target values in optimization. This dual formulation corresponds to the following cost function, which can be directly compared with our cross entropy and (19):

$$J(W_1, W_2) = \sum_{x_t, y_t \in X} (1 - r_t f_{\{w_1, w_2\}}(x_t, y_t))$$  \hspace{1cm} (23)

Hu et al. proposed the discriminative deep metric learning (DDML) approach, which is also based on Siamese neural networks [39]. This work proposes taking image representations to another feature subspace, by applying shared neural network operations and then calculating the Euclidean distance on this new space:

$$d^2_I(x, y) = ||f(x) - f(y)||^2_2$$  \hspace{1cm} (24)

where,

$$h^{(1)} = \sigma(W^{(1)} x + b^{(1)}) \hspace{1cm} h^{(2)} = \sigma(W^{(2)} h^{(1)} + b^{(2)}) \hspace{1cm} \ldots \hspace{1cm} f(x) = h^{(M)} = \sigma(W^{(M)} h^{(M-1)} + b^{(M)})$$  \hspace{1cm} (25)

The distance value calculated in the new space is used to enforce a margin between friends and foes via a threshold value. The pairs that do not conform to this margin are designed to contribute to the cost. As the cost function, the generalized logistic loss function is used, which is a smoothed approximation of the rectified linear unit function [40].

$$J = \sum_{t} \log \left(1 + exp \left(\beta(1 - r_t (\tau - d^2_I(x_t, y_t)))\right)\right)$$

$$+ \lambda \sum_{m} (||W^{(m)}||^2_2 + ||h^{(m)}||^2_2)$$  \hspace{1cm} (26)

In (26), the first term is the generalized logistic function, steepness of which is decided by $\beta$. The part inside the exponential is contributed to the cost when it is non-zero. This means that a margin ($\tau$) is forced to be kept between the distances of friend and foe pairs. One point to make here is that although DDML, like our study, use the term "metric", $d_I(x, y)$ is not guaranteed to conform to the axioms of metric spaces, on the space that contains $x$ and $y$. Such "metric" learning approaches are, in fact, representation or space learning approaches such that the target distance measure performs better on the new feature space.

Zheng et al. proposed the logistic similarity learning approach [21], which also aims to learn a shared projection matrix to optimize over the learned cosine similarity (18), similar to [19]. The interesting part is that they use generalized logistic loss function, given by (26), as in [39], which penalizes for wrong signed cosine similarities of pairs.

$$J = \sum_{t} \log \left(1 + exp \left( - \frac{r_t(CS(x_t, y_t, W) - b)}{T} \right)\right)$$

$$+ \lambda ||W - W_o||^2_2$$  \hspace{1cm} (27)

The positive constants $b$ and $T$ are used to shift the decision boundary and control the steepness, respectively. Although the original works adopt very different techniques, their objective functions depicted by (26) and (27) are directly comparable. The former aims to maximize the gap between Euclidean distance measures of friend and foe pairs, while the latter aims to push the cosine distances of friends and...
Two very recent studies have worked on the idea of jointly training a set of feature vectors for a common similarity metric. Lu et al. extended their earlier DDML work, to include the joint training of different features [13] and called it discriminative deep multi-metric learning (DDMML). The function \( f \) in (25) is now obtained for \( K \) different feature representations. If we call the \( k^{th} \) feature representation of an image \( x^k \), each DDML learns a feature space representation \( h^k \theta_k \), and a distance measure \( d_{jk}(x^k, y^k) \), where \( M \) and \( k \) denote layer depth and feature representation, respectively. Each DDML in DDMML also learns a weight parameter \( a_k \) for each feature. The joint distance is defined as:

\[
d^j_k(x,y) = \sum_{k=1}^{K} a_k d^2_{jk}(x^k,y^k) = \sum_{k=1}^{K} a_k \| f_k(x) - f_k(y) \|_2^2
\]

The cost function of DDMML involves the weighted sum of each DDML cost function, given with (26), plus a penalty term calculated with the squared differences of \( d_{jk} \)'s for the same image. The accordance of different feature representations for the same image is enforced with this term.

Another very relevant work addresses the problem of joint similarity learning as a multi-view metric learning [28]. Similar to DDMML, they define a joint distance measure obtained by the summation of individual and shared neural network activations. Given a pair of different representations of the same image, \( x^a \) and \( x^c \), two projection matrices are defined:

- \( W_k^s \): specific projection matrix for feature \( k \), symmetric positive definite
- \( W_{k,\ell}^s \): common projection matrix for features \( k \) and \( \ell \), symmetric positive definite

The total distance is then defined as:

\[
d^a_k(x,y) = \sum_{k=1}^{K} (x_k - y_k)^T W^s_k (x_k - y_k) + \sum_{k=1}^{K} \sum_{\ell=1}^{L} (x_k - y_\ell)^T W^c_{k,\ell} (x_k - y_\ell)
\]

The methodologies of [13] and [28] can be compared to our work in that all three works suggest a joint learning of similarity metric, for multiple features. While these two studies work on finding neural network-based representations for Euclidean space, our proposed approach incorporates this idea into the inner product similarity-based realm.

4 Experiments

The experiments are conducted on the YTF dataset and the view-2 configuration of the LFW dataset. YTF dataset consists of video snippets of that are split to frames at 24 frames per second. It was ensured prior to the dataset creation that the videos are not still-image slide shows and there are no identical videos. Each image in the video is re-scaled and variants of LBP features are extracted, including Center-Symmetric LBP (CSLBP) and Four-Patch LBP. We took the LBP and FLBP features in CoSiM training in this work. The verification task is set with the randomly collected 5,000 video pairs from the database. These videos are divided into 10 splits, each containing 250 friends and foes each. The goal of the task is to determine if the videos in a pair is friends or foes.

Hence, in order to match the first set-up of image verification we took one random frame from each of the videos provided in the database and conducted CoSiM experiments on these images. The results reported with this dataset strictly follows the restricted protocol, since no additional data is used in either feature extraction or distance metric training.

The LFW dataset includes 13,233 images that belong to 5,749 different people, which can be viewed and downloaded at http://vis-www.cs.umass.edu/lfw/. One aspect that makes this dataset difficult in machine learning tasks is that more than 70% of individuals in the dataset have only one image. A second difficulty is that the people in the training-test split are mutually exclusive. In other words, none of the people that are seen in the training set exist in the test set. This issue is referred to as the unseen pair matching problem. Sample image pairs from the dataset can be seen in Figure 6. It can be observed that two pictures of the same person can possess a considerable degree of variation whereas pictures of different people may even look more similar than those of same people.
We train the SML network with each of two features and compare the verification performance using the three distance/similarity metrics, i.e. cosine, Euclidean and SML. Table 4 shows the mean accuracy and the standard deviation over the 10-fold cross validation. The performance of verification when the CoSiM similarity values are used as the discrimination criteria is also calculated. It can be observed that the similarity metric learning provides about 3% absolute improvement in the validation accuracy. Furthermore, the validation performance is increased for another 2% absolute, when the similarity is learned collaboratively using the two features. The effects of the proposed systems are visualized in Figure 7.

Table 4 Experimental Results on YTF Dataset

<table>
<thead>
<tr>
<th>Feature</th>
<th>Distance</th>
<th>Accuracy</th>
</tr>
</thead>
<tbody>
<tr>
<td>FPLBP</td>
<td>cosine</td>
<td>61.94 ± 1.23</td>
</tr>
<tr>
<td>FPLBP</td>
<td>Euclidean</td>
<td>62.32 ± 1.08</td>
</tr>
<tr>
<td>LBP</td>
<td>cosine</td>
<td>62.72 ± 1.41</td>
</tr>
<tr>
<td>LBP</td>
<td>Euclidean</td>
<td>62.91 ± 1.36</td>
</tr>
<tr>
<td>LBP</td>
<td>SML</td>
<td>65.88 ± 1.86</td>
</tr>
<tr>
<td>FPLBP</td>
<td>SML</td>
<td>68.04 ± 3.03</td>
</tr>
<tr>
<td>both feats collaboratively CoSiM-m</td>
<td>CoSiM-m</td>
<td>69.92 ± 2.50</td>
</tr>
<tr>
<td>both feats collaboratively CoSiM-a</td>
<td>CoSiM-a</td>
<td>71.27 ± 2.48</td>
</tr>
</tbody>
</table>

![Fig. 7: Effect of SML and CoSiM on validation accuracy, compared with other distance metrics on YFT dataset.](image)

4.2.2 Experiments on LFW: In this dataset we also evaluate the performance of our system in comparison with a very similar method of LDML [7]. We see that both SML-S and SML-L outperform this baseline which use SIFT features as input and the similarity metric given in (16) where it was absolute, i.e. sigma distance with SIFT features. Table 4 shows the mean accuracy and the standard deviation over the 10-fold cross validation.

4.2.3 Discriminative Power of CoSiM: As another interrogation of the proposed approach we compared our similarity learning approach with the two most widely used distance metrics, i.e. the cosine distance (17) and the Euclidean distance (30). The distance equivalent of the sigma similarity in
The Institution of Engineering and Technology 2019 9
IET Research Journals, pp. 1–11

(4), is obtained by calculating the complement since it can be interpreted as a probability measure [36],

\[ d_x = 1 - \sigma(x^TW^TW + b). \]  

Similarly, the collaborative distance values that make use of two features are calculated with the CoSiM similarity values given in (4) and (8):

\[ d_{CoSiM} = 1 - f(x_1, x_2, y_1, y_2) \]  

For this set of experiments, we calculated the distances between the pairs of friends and the pairs of foes in the test datasets. We calculated the histograms of distance values between 3,000 friends and 3,000 foes in the LFW dataset, using different image representations and distance metrics. It should be noted that the test images are strictly excluded for pertinent training subset for the learning-based distance metrics. In other words, we train a new network from scratch for each of the subsets. Figure 8 exhibits the histograms of friend and foe distances, calculated with Euclidean distance, cosine distance and the sigma distance.

The means of the distances between friends and foes (i.e. dispersion statistics) can be seen in Table 6. It can be seen both on the table and Figure 8 that the Euclidean distance and the cosine distance perform poorly on discriminating friends and foes.

Table 6 Dispersion statistics of different distance metrics.

<table>
<thead>
<tr>
<th>DISTANCE CALCULATION</th>
<th>DISPERSION BETWEEN</th>
</tr>
</thead>
<tbody>
<tr>
<td>representation</td>
<td>friends</td>
</tr>
<tr>
<td>SIFT (30)</td>
<td>21.757</td>
</tr>
<tr>
<td>LBP (30)</td>
<td>19.180</td>
</tr>
<tr>
<td>SIFT (17)</td>
<td>0.859</td>
</tr>
<tr>
<td>LBP (17)</td>
<td>0.837</td>
</tr>
<tr>
<td>SIFT (31)</td>
<td>0.376</td>
</tr>
<tr>
<td>LBP (31)</td>
<td>0.389</td>
</tr>
<tr>
<td>CoSiM-m (32)</td>
<td>0.309</td>
</tr>
<tr>
<td>CoSiM-a (32)</td>
<td>0.299</td>
</tr>
</tbody>
</table>

The mean distance values show that the learned distance metrics help discriminate friends and foes better than the existing metrics. Furthermore, the proposed collaborative distance metrics provide a better discrimination than the single feature-based ones. The histogram plots of the distances calculated with the CoSiM methods and late fusion (c) of the sigma distance-based systems is given in Figure 9. The comparison of (a and b) to (c) shows how collaborative learning is more efficient than learning and then fusing.

5 Discussion and Conclusion

In this paper, a joint Siamese neural network training methodology is proposed for exploiting multiple face image features. The training methodology we refer to as CoSiM is an extension of the sigma distance metric learning network that was previously applied for speech features in a keyword search task. It was shown that the SML technique also works for image features such as SIFT and LBP and furthermore, the CoSiM methodology acts as an effective means of early fusion of several features. The experiments conducted on the YTF and LFW dataset show that, the results of the CoSiM training performs better than late fusion of the individual distance metrics. The performance increase obtained by combining different approaches demonstrates the complementary nature of our approach. We see that SML-based methods enhanced with the joint training setup yield competitive results compared to other works in literature. The methods which outperform the proposed method generally use significantly deeper architectures or more sophisticated...
Fig. 9: Normalized histograms of dispersion between *friends* (blue) and *foes* (orange) as a comparison of late vs early fusion techniques.

### Table 7: Comparison with the Literature

<table>
<thead>
<tr>
<th>Method</th>
<th>Accuracy</th>
</tr>
</thead>
<tbody>
<tr>
<td>MERL [2]</td>
<td>70.52 ± 0.60</td>
</tr>
<tr>
<td>MERL+Nowak [2]</td>
<td>76.18 ± 0.58</td>
</tr>
<tr>
<td>LDM [7]</td>
<td>79.27 ± 0.60</td>
</tr>
<tr>
<td>NReLU [8]</td>
<td>80.73 ± 1.34</td>
</tr>
<tr>
<td>Single LE + holistic [5]</td>
<td>81.22 ± 0.53</td>
</tr>
<tr>
<td>DML-eig SIFT [44]</td>
<td>81.27 ± 2.30</td>
</tr>
<tr>
<td>Hybrid, aligned [3]</td>
<td>83.98 ± 0.35</td>
</tr>
<tr>
<td>LARK supervised [3]</td>
<td>85.10 ± 0.59</td>
</tr>
<tr>
<td>LBP + CSML [19]</td>
<td>85.57 ± 0.52</td>
</tr>
<tr>
<td>Hybrid on LFW3D [45]</td>
<td>85.63 ± 0.53</td>
</tr>
<tr>
<td>DML-eig combined [44]</td>
<td>85.65 ± 0.56</td>
</tr>
<tr>
<td>Combined b/g samples [4]</td>
<td>86.83 ± 0.34</td>
</tr>
<tr>
<td>TSMWL with OCLBP [9]</td>
<td>87.10 ± 0.43</td>
</tr>
<tr>
<td>VFV [41]</td>
<td>87.47 ± 1.49</td>
</tr>
<tr>
<td>Pose Adaptive Filter (PAF) [46]</td>
<td>87.77 ± 0.51</td>
</tr>
<tr>
<td>Convolutional DBN [47]</td>
<td>87.77 ± 0.62</td>
</tr>
<tr>
<td>CSML + SVM [19]</td>
<td>88.00 ± 0.37</td>
</tr>
<tr>
<td>HT Brain-Inspired Feat.s [6]</td>
<td>88.13 ± 0.58</td>
</tr>
<tr>
<td>SFRD+PMMAL [48]</td>
<td>89.35 ± 0.50</td>
</tr>
<tr>
<td>LML3 [49]</td>
<td>89.57 ± 0.43</td>
</tr>
<tr>
<td>Spartans [50]</td>
<td>89.69 ± 0.36</td>
</tr>
<tr>
<td>Sub-SML [10]</td>
<td>89.73 ± 0.38</td>
</tr>
<tr>
<td>TSMWL with feature fusion [9]</td>
<td>89.80 ± 0.47</td>
</tr>
<tr>
<td>DDML [39]</td>
<td>90.68 ± 1.41</td>
</tr>
<tr>
<td>Sub-SML + Hybrid on LFW3D [45]</td>
<td>91.65 ± 1.04</td>
</tr>
<tr>
<td>HPEN + HD-LBP + DDML [51]</td>
<td>92.57 ± 0.36</td>
</tr>
<tr>
<td><strong>this work (JSML+VFV+Sub-SML)</strong></td>
<td><strong>92.64 ± 0.86</strong></td>
</tr>
<tr>
<td>HPEN + HD-Gabor + DDML [51]</td>
<td>92.80 ± 0.47</td>
</tr>
<tr>
<td>MSBSIF-SIEDA [52]</td>
<td>94.63 ± 0.95</td>
</tr>
</tbody>
</table>

**feature representations.**

The experimental set-up for this paper was constrained to using two features per network. However, it should be noted that the methodology can very well be extended to multiple features and the addition of more features can be quantified as a future study. Also, we used hand crafted features in this paper, on the Siamese ends of the networks. The feed-forward nature of the network’s two sides can be changed into CNN networks and the method can be made end-to-end.

### 6 Acknowledgments

Authors would like to thank the anonymous reviewers of this paper for their invaluable comments that made the work better.

### 7 References


