

BioMetricNet: deep unconstrained face verification through learning of metrics regularized onto Gaussian distributions

*Original*

BioMetricNet: deep unconstrained face verification through learning of metrics regularized onto Gaussian distributions / Ali, A., Testa, M., Bianchi, T., Magli, E.. - STAMPA. - (2020), pp. 133-149. (European Conference on Computer Vision Glasgow ) [10.1007/978-3-030-58595-2\_9].

*Availability:*

This version is available at: 11583/2846483 since: 2020-09-23T10:39:58Z

*Publisher:*

Springer

*Published*

DOI:10.1007/978-3-030-58595-2\_9

*Terms of use:*

This article is made available under terms and conditions as specified in the corresponding bibliographic description in the repository

*Publisher copyright*

Springer postprint/Author's Accepted Manuscript

This version of the article has been accepted for publication, after peer review (when applicable) and is subject to Springer Nature's AM terms of use, but is not the Version of Record and does not reflect post-acceptance improvements, or any corrections. The Version of Record is available online at: [http://dx.doi.org/10.1007/978-3-030-58595-2\\_9](http://dx.doi.org/10.1007/978-3-030-58595-2_9)

(Article begins on next page)

# BioMetricNet: deep unconstrained face verification through learning of metrics regularized onto Gaussian distributions

Arslan Ali<sup>[0000-0003-0282-0726]</sup>, Matteo Testa<sup>[0000-0003-2628-6433]</sup>,  
Tiziano Bianchi<sup>[0000-0002-3965-3522]</sup>, and Enrico Magli<sup>[0000-0002-0901-0251]</sup>

Department of Electronics and Telecommunications  
Politecnico di Torino, Italy  
{arslan.ali, matteo.testa, tiziano.bianchi, enrico.magli}@polito.it

**Abstract.** We present BioMetricNet: a novel framework for deep unconstrained face verification which learns a regularized metric to compare facial features. Differently from popular methods such as FaceNet, the proposed approach does not impose any specific metric on facial features; instead, it shapes the decision space by learning a latent representation in which matching and non-matching pairs are mapped onto clearly separated and well-behaved target distributions. In particular, the network jointly learns the best feature representation, and the best metric that follows the target distributions, to be used to discriminate face images. In this paper we present this general framework, first of its kind for facial verification, and tailor it to Gaussian distributions. This choice enables the use of a simple linear decision boundary that can be tuned to achieve the desired trade-off between false alarm and genuine acceptance rate, and leads to a loss function that can be written in closed form. Extensive analysis and experimentation on publicly available datasets such as Labeled Faces in the wild (LFW), Youtube faces (YTF), Celebrities in Frontal-Profile in the Wild (CFP), and challenging datasets like cross-age LFW (CALFW), cross-pose LFW (CPLFW), In-the-wild Age Dataset (AgeDB) show a significant performance improvement and confirms the effectiveness and superiority of BioMetricNet over existing state-of-the-art methods.

**Keywords:** Biometrics, face verification, biometric authentication

## 1 Introduction

Over the last few years, huge progress has been made in the deep learning community. Advances in convolutional neural networks (CNN) have led to unprecedented accuracy in many computer vision tasks. One of those that have attracted computer vision researchers since its inception is being able to recognize a person from a picture of their face. This task, which has countless applications, is still far to be marked as a solved problem. Given a pair of (properly aligned) face images, the goal is to make a decision on whether they represent the same person or not.

Early attempts in the field required the design of handcrafted features that could capture the most significant traits that are unique to each person. Furthermore, they had to be computed from a precisely aligned and illumination normalized picture. The complexity of handling the non-linear variations that may occur in face images later became evident, and explained the fact that those methods tend to fail in non-ideal conditions.

A breakthrough was then made possible by employing features learned through CNN-based networks, e.g. DeepFace [1] and DeepID [2]. As in previous methods, once the features of two test faces have been computed, a distance measure (typically  $\ell_2$  norm) is employed for the verification task: if the distance is below a certain threshold the two test faces are classified as belonging to the same person, otherwise not. The loss employed to compute such features is the softmax cross-entropy. Indeed, it was found that the generalization ability could be improved by maximizing inter-class variance and minimizing intra-class variance. Works such as [3, 4] adopted this strategy by accounting for a large margin, in the Euclidean space, between “contrastive” embeddings. A further advance was then brought by FaceNet [5] which introduced the triplet-loss, whereby the distance between the embeddings is evaluated in relative rather than absolute terms. The introduction of the anchor samples in the training process allows to learn embeddings for which the anchor-positive distance is minimized while the anchor-negative distance is maximized. Even though this latter work has led to better embeddings, it has been shown that it is oftentimes complex to train [6]. The focus eventually shifted to the design of new architectures employing metrics other than  $\ell_2$  norm to provide more strict margins. In [7] and [8] the authors propose to use angular distance metrics to enforce a large margin between negative examples and thus reduce the number of false positives.

In all of the above-mentioned methods, a pre-determined analytical metric is used to compute the distance between two embeddings, and the loss function is designed in order to ensure a large margin (in terms of the employed metric) among the features of negative pairs while compacting the distance among the positive ones. It is important to underline that the chosen metric is a critical aspect in the design of such neural networks. Indeed, a large performance increase has been achieved with the shift from Euclidean to angular distance metrics [9, 10].

In this work we propose a different approach: not only we aim to learn the most discriminative features, but also to jointly learn the best (possibly highly non-linear) metric to compare such features. The only requirement we impose determines how the metric should behave depending on whether the features are coming from matching or non-matching pairs. Specifically, we regularize the metric output such that its values follow two different statistical distributions: one for matching pairs, and the other for non-matching pairs (see Fig. 1).

The idea of relying on the (empirical) distributions of the feature distances in order to improve their discriminative ability was discussed in [11]. In the above work, the authors introduce the histogram loss in order to minimize the overlap between the histograms of the distances of matching and non-matching feature

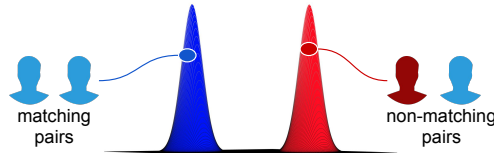


Fig. 1: The goal of BioMetricNet is to map the input pairs onto target distributions in the latent space. Matching pairs (same user - blue) are mapped to a target distribution whose mean value is far from that of the non-matching pairs (different users - red).

pairs, so as to obtain more regularized features. However, while this approach fits well clustering tasks in which one is only interested in relative distances between pairs, it is not suited for the verification problem we are considering in this paper: the decision boundary between the two histograms is highly dependent on the employed dataset and does not generalize across different data distributions. The approach we follow is rather different: by regularizing a latent space by means of target distributions we *impose* the desired shape (based on a possibly highly non-linear metric) and thus have a known and fixed decision boundary which generalizes across different datasets. This seminal idea of employing target distributions was first introduced in [12] and [13]. However, it is important to underline that in [12] and [13] it was used to solve a one-against-all classification problem, regularizing a latent space such that the biometric traits of *a single user* would be mapped onto a distribution, and those of every other possible user onto another distribution, so that a thresholding decision could be used to identify biometric traits belonging to that specific user. The above methods also required a user-specific training of the neural network.

Conversely, besides learning features, the neural network proposed in this paper, which we name BioMetricNet, shapes the decision *metric* such that pairs of similar faces are mapped to a distribution, whereas pair of dissimilar faces are mapped to a different distribution, thereby avoiding user-specific training. This approach has several advantages: i) Since the distributions are known, and generally simple, the decision boundaries are simple, too. This is in contrast with the typical behavior of neural networks, which tend to yield very complex boundaries; ii) If the distributions are taken as Gaussian with the same variance, then a hyperplane is the optimal decision boundary. This leads to a very simple classifier, which learns a complex mapping to a high-dimensional latent space, in a way that mimics kernel-based methods. Moreover, Gaussian distributions are amenable to writing the loss function in closed form; iii) Mapping to known distributions easily enables to obtain confidences for each test sample, as more difficult pairs are mapped to the tails of the distributions. Since in BioMetricNet the distribution of the metric output values is known, the decision threshold can be tuned to achieve the desired level of false alarm rate or genuine acceptance rate.

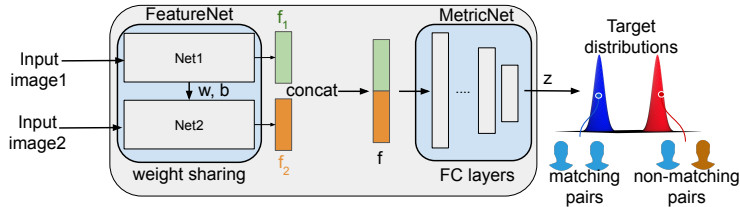


Fig. 2: BioMetricNet architecture during the training phase. After face detection and alignment, matching and non-matching face pairs are given as an input to the FeatureNet to extract the discriminative face features from the image space  $\mathbf{x}$  into feature vector space  $\mathbf{f}_i \in \mathbb{R}^d$ . The feature vectors are concatenated  $\mathbf{f} = [\mathbf{f}_1 \mathbf{f}_2] \in \mathbb{R}^{2d}$  and passed to the MetricNet which maps  $\mathbf{f}$  onto well-behaved target distributions  $\mathbf{z} \in \mathbb{R}^p$  in the latent space.

The resulting design, employing the best learned metric for the task at hand, allows us to improve over the state-of-the-art also in the case of very challenging datasets, as will be shown in Sec. 3.7. We stress that, although in this paper BioMetricNet is applied to faces, the method is general and can be applied to other biometric traits or data types; this is left as future work.

## 2 Proposed Method

BioMetricNet strives to learn meaningful features of the input faces along with a discriminative metric to be used to compare two sets of facial features. More specifically, as depicted in Fig. 2, BioMetricNet is made of two sub-networks: FeatureNet and MetricNet. The former is a siamese network which processes pairs of input faces  $\mathbf{x} = [\mathbf{x}_1, \mathbf{x}_2]$  and outputs a pair of facial features  $\mathbf{f} = [\mathbf{f}_1, \mathbf{f}_2]$  for both matching and non-matching input pairs. MetricNet is then employed to map these feature pairs onto a point  $\mathbf{z}$  in a  $p$ -dimensional space in which a decision is made. These two networks are trained as a single entity to match the desired behavior. Their architecture is described more in detail in Sec. 2.1.

The novelty of our approach is that we do not impose any predetermined metric between  $\mathbf{f}_1$  and  $\mathbf{f}_2$ : the metric is rather learned by MetricNet shaping the decision space according to two target distributions through the loss function, as described in the following. The loss function forces the value of the learned metric to follow different statistical distributions when applied to matching and non-matching pairs, respectively. Although arbitrary target distributions can be employed, a natural choice is to use distributions that have far-enough mass centers, lead to simple decision boundaries, and lend themselves to writing the loss function in a closed form.

For BioMetricNet, let us denote as  $\mathbb{P}_m$  and  $\mathbb{P}_n$  the desired target distributions for matching and non-matching pairs, respectively. We choose  $\mathbb{P}_m$  and  $\mathbb{P}_n$  to be

multivariate Gaussian distributions over a  $p$ -dimensional space:

$$\mathbb{P}_m = \mathcal{N}(\boldsymbol{\mu}_m, \boldsymbol{\Sigma}_m), \mathbb{P}_n = \mathcal{N}(\boldsymbol{\mu}_n, \boldsymbol{\Sigma}_n), \quad (1)$$

where  $\boldsymbol{\Sigma}_m = \sigma_m^2 \mathbb{I}_p$  and  $\boldsymbol{\Sigma}_n = \sigma_n^2 \mathbb{I}_p$  are diagonal covariance matrices and  $\boldsymbol{\mu}_m = \mu_m \mathbf{1}_p^T$ ,  $\boldsymbol{\mu}_n = \mu_n \mathbf{1}_p^T$  are the expected values. The choice of using Gaussian distributions is a very natural one in this context. Because of the central limit theorem [14], the output of fully connected layers tends to be Gaussian distributed. Moreover, if  $\boldsymbol{\Sigma}_m = \boldsymbol{\Sigma}_n$ , then a linear decision boundary (hyperplane) is optimal for this Gaussian discrimination problem. Therefore, while in general BioMetricNet can be trained to match arbitrary distributions, in the following we will describe this specific case. It can also be noted that using different variance for the two distributions would complicate the choice of the parameters, since the optimal variance will be specific to the considered dataset in order to match its intra and inter-class variances.

As said above, in the Gaussian case the loss function can be written in closed form. Let us define  $\mathbf{x}_m$  and  $\mathbf{x}_n$  as the pairs of matching and non matching face images, respectively. In the same way we define  $\mathbf{f}_m$  and  $\mathbf{f}_n$  as the corresponding features output by FeatureNet. MetricNet can be seen as a generic encoding function  $H(\cdot)$  of the input feature pairs  $\mathbf{z} = H(\mathbf{f})$ , where  $\mathbf{z} \in \mathbb{R}^p$ , such that  $\mathbf{z}_m \sim \mathbb{P}_m$  if  $\mathbf{f} = \mathbf{f}_m$  and  $\mathbf{z}_n \sim \mathbb{P}_n$  if  $\mathbf{f} = \mathbf{f}_n$ . As previously described, we want to regularize the metric space where the latent representations  $\mathbf{z}$  lie in order to constrain the metric behavior. Since the distributions we want to impose are Gaussian, the Kullback-Leibler (KL) divergence between the sample and target distributions can be obtained in closed-form as a function of only first and second order statistics and can be easily minimized. More specifically, the KL divergence for multivariate Gaussian distributions can be written as:

$$\mathcal{L}_m = \frac{1}{2} \left[ \log \frac{|\boldsymbol{\Sigma}_m|}{|\boldsymbol{\Sigma}_{S_m}|} - p + \text{tr}(\boldsymbol{\Sigma}_m^{-1} \boldsymbol{\Sigma}_{S_m}) + (\boldsymbol{\mu}_m - \boldsymbol{\mu}_{S_m})^\top \boldsymbol{\Sigma}_m^{-1} (\boldsymbol{\mu}_m - \boldsymbol{\mu}_{S_m}) \right]. \quad (2)$$

where the subscript  $S$  indicates the sample statistics.

Interestingly, since we only need the first and second order statistics of  $\mathbf{z}$ , we can capture this information batch-wise. As will be explained in detail in Sec. 2.2, during the training the network is given as input a set of face pairs from which a subset of  $b/2$  difficult matching and  $b/2$  difficult non-matching face pairs are extracted, being  $b$  the batch size. Letting  $\mathbf{X} \in \mathbb{R}^{b \times r}$  with  $r$  the size of a face pair, this results in a collection of latent space points  $\mathbf{Z} \in \mathbb{R}^{b \times p}$  after the encoding. We thus compute first and second order statistics of the encoded representations  $\mathbf{Z}_m, \mathbf{Z}_n$  related to matching  $(\boldsymbol{\mu}_{S_m}, \boldsymbol{\Sigma}_{S_m})$  and non-matching  $(\boldsymbol{\mu}_{S_n}, \boldsymbol{\Sigma}_{S_n})$  input faces respectively. More in detail, let us denote as  $\boldsymbol{\Sigma}_{S_m}^{(ii)}$  the  $i$ -th diagonal entry of the sample covariance matrix of  $\mathbf{Z}_m$ . The diagonal covariance assumption allows us to further simplify (2) as:

$$\mathcal{L}_m = \frac{1}{2} \left[ \log \frac{\sigma_m^{2p}}{\prod_i \boldsymbol{\Sigma}_{S_m}^{(ii)}} - p + \frac{\sum_i \boldsymbol{\Sigma}_{S_m}^{(ii)}}{\sigma_m^2} + \frac{\|\boldsymbol{\mu}_m - \boldsymbol{\mu}_{S_m}\|_2}{\sigma_m^2} \right]. \quad (3)$$

This loss captures the statistics of the matching pairs and enforces the target distribution  $\mathbb{P}_m$ . For brevity we omit the derivation of  $\mathcal{L}_n$  which is obtained similarly.

Then, the overall loss function which will be minimized end-to-end across the whole network (FeatureNet and MetricNet) is given by  $\mathcal{L} = \mathcal{L}_m + \mathcal{L}_n$ .

## 2.1 Architecture

In the following we discuss the architecture and implementation strategy of FeatureNet and MetricNet.

**FeatureNet** The goal of FeatureNet is to extract the most distinctive facial features from the input pairs. The architectural design of FeatureNet is crucial. In general, one may employ any state-of-the-art neural network architecture able to learn good features. Due to its fast convergence, in our tests we employ a siamese Inception-ResNet-V1 [15]. The output size of the stem block in Inception-ResNet is  $35 \times 35 \times 256$ , followed by 5 blocks of Inception-ResNet-A, 10 blocks of Inception-ResNet-B and 5 blocks of Inception-ResNet-C. At the bottom of the network we employ a fully connected layer with output size equal to the feature vector dimensionality  $d$ . The employed dropout rate is 0.8. The pairs of feature vectors  $\mathbf{f}_1$  and  $\mathbf{f}_2$  in output of FeatureNet are concatenated resulting in  $\mathbf{f} = [\mathbf{f}_1 \mathbf{f}_2] \in \mathbb{R}^{2d}$  and given as input to MetricNet.

**MetricNet** The goal of MetricNet is to learn the best metric based on feature vector  $\mathbf{f}$  and to map it onto the target distributions in the latent space. MetricNet consists of 7 fully connected layers with ReLU activation functions at the output of each layer. At the last layer, no activation function is employed. The input size of MetricNet is  $2d$ , the first fully connected layer has an output size equal to  $2d$ , the output size keeps decreasing gradually by a factor of 2 with the final layer having an output size equal to the latent space dimensionality  $p$ .

We also highlight that MetricNet, by taking as input  $\mathbf{f} = [\mathbf{f}_1, \mathbf{f}_2]$ , allows us to model any arbitrary nonlinear correlations between the feature vectors. Indeed, the use of an arbitrary combination of the input features entries has been proven to be highly effective, see e.g. [16, 17].

## 2.2 Pairs Selection during Training

For improved convergence, BioMetricNet selects the most difficult matching and non-matching pairs during training, i.e., those far from the mean values of the target distributions and close to the threshold. For each mini-batch, at the end of the forward pass we select the subset of matching pairs whose output  $\mathbf{z}_m$  is sufficiently distant from the mass center of  $\mathbb{P}_m$ , i.e.  $\|\mathbf{z}_m - \boldsymbol{\mu}_m\|_\infty \geq 2\sigma_m$ . Similarly, for the non-matching pairs we select those which result in a  $\mathbf{z}_n$  such that  $\|\mathbf{z}_n - \boldsymbol{\mu}_n\|_\infty \geq 2\sigma_n$ . Then, in the backward pass we minimize the loss over a subset of  $b/2$  difficult matching and  $b/2$  difficult non-matching pairs with  $b$

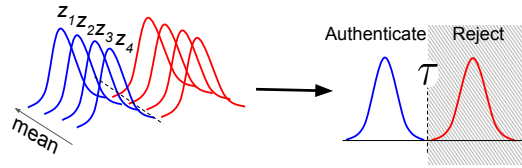


Fig. 3: During the testing phase, we obtain the latent vectors of the input image pair and its three horizontal flips. For all experiments, the final latent space vector is calculated as  $\bar{\mathbf{z}} = \frac{1}{4} \left( \sum_{i=1}^4 \mathbf{z}_i \right)$ . Pairs are classified as matching and non-matching by comparing  $\bar{\mathbf{z}}$  with a threshold  $\tau$ .

being the mini-batch size of the selected difficult pairs. In order to have a stable training, the backward pass is executed only when we are able to collect  $b/2$  difficult matching and  $b/2$  difficult non-matching pairs, else the mini-batch is discarded.

The rationale behind this choice comes from the result of the latent space regularization. Indeed, as one traverses the latent space from  $\boldsymbol{\mu}_m$  towards  $\boldsymbol{\mu}_n$ , one moves from very similar face pairs to very dissimilar ones. Points close to the threshold can be thought of as representing pairs for which the matching/non-matching uncertainty is high. As the training proceeds, at every new epoch the network improves the mapping of the “difficult” pairs as it is trained on pairs for which it is more difficult to determine whether they represent a match or not.

### 2.3 Authentication

In the testing phase, a pair of images are passed through the whole network in order to compute the related metric value  $\mathbf{z}$ . Then, a decision is made according to this value. As said, for our choice of target distributions a hyperplane can be used for the optimal decision, i.e., we can use the test

$$(\boldsymbol{\mu}_m - \boldsymbol{\mu}_n)^T \mathbf{z} \leq (\boldsymbol{\mu}_m - \boldsymbol{\mu}_n)^T (\boldsymbol{\mu}_m + \boldsymbol{\mu}_n) / 2. \quad (4)$$

For  $p = 1$ , this boils down to comparing the scalar  $\mathbf{z}$  with a threshold  $\tau = (\mu_m + \mu_n) / 2$ .

However, we consider an improved approach which is able to capture additional information: we use flipped images to compute supplementary features as done in the recent literature [8, 10]. Namely, given an image pair, we compute the metric output  $\mathbf{z}$  for the original image pair as well as the 3 pairs resulting from the possible combinations of horizontally flipped and non-flipped images. We employ a horizontal flip defined as  $(x, y) \rightarrow (\text{width} - x - 1, y)$ . We thus obtain 4 metric values. Then, the decision is performed on a value  $\bar{\mathbf{z}} = \frac{1}{4} \left( \sum_{i=1}^4 \mathbf{z}_i \right)$ , where  $\mathbf{z}_i$  is the metric output corresponding to the  $i$ -th image flip combination, see Fig. 3. The expected value of  $\bar{\mathbf{z}}$  in case of matching and non-matching pairs is still equal to  $\boldsymbol{\mu}_m$  and  $\boldsymbol{\mu}_n$ , respectively. Therefore, the test (4) will still be valid

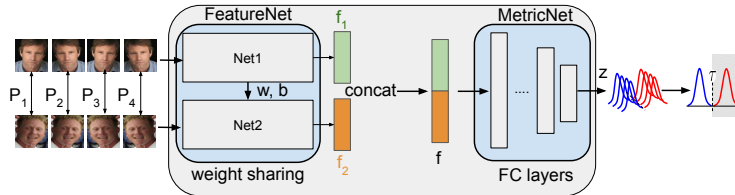


Fig. 4: BioMetricNet architecture during the testing phase. Given a pair of images to be tested, after face detection and alignment, by accounting for all the possible horizontal flip combinations, we obtain 4 image pairs, i.e.  $P_1, P_2, P_3$  and  $P_4$ . The latent vectors of the corresponding pairs are computed and aggregated to  $\bar{z}$  and compared with a threshold  $\tau$ .

on  $\bar{z}$ . Fig. 4 depicts BioMetricNet during the authentication phase.  $P_1$  represents the input image pair, and  $P_2, P_3$ , and  $P_4$  represent the three horizontal flips.

### 3 Experiments

#### 3.1 Experimental Settings

The network is trained with Adam optimizer [18] using stochastic gradient descent [19, 20]. Each epoch consists of 720 people with each person having a minimum of 5 images to ensure enough matching and non-matching pairs. We set the batch size  $b$  to 220 difficult pairs to obtain statistically significant first and second order statistics. Each batch is balanced, i.e. it contains half matching and half non-matching pairs. The initial learning rate is set to 0.01 with an exponential decay factor of 0.98 after every 5 epochs. In total, the network is trained for 500000 iterations. Weight decay is set to  $2 \times 10^{-4}$ . We further employ dropout with a keep probability value equal to 0.8. All experiments are implemented in TensorFlow [21]. For the augmentation horizontal flips of the images are taken.

#### 3.2 Preprocessing

For preprocessing we follow the strategies adopted by most recent papers in the field [9, 10, 8]. For both the training and testing datasets, we employ MTCNN [22] to generate normalized facial crops of size  $160 \times 160$  with face alignment based on five facial points. As a final step, the images are mean normalized and constrained in the range  $[-1, 1]$  as done in [10, 8, 9].

#### 3.3 Datasets

**Training** The training datasets are those commonly used in recent works in the field. More in detail, we use different datasets for training and testing phases. The datasets we employ during training are Casia [23] (0.49M images having 10k identities) and MS1M-DeepGlint (3.9M images having 87k identities) [24].

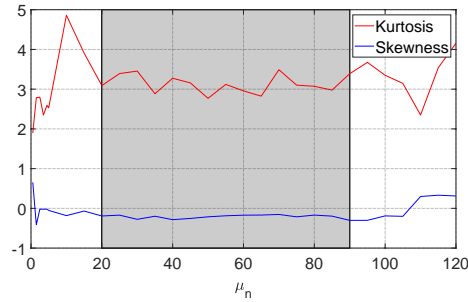


Fig. 5: Kurtosis and skewness of the latent space metric on LFW when  $\mathbb{P}_n = \mathcal{N}(w, 1)$  where  $w = [0.5, 120]$ , and  $\mathbb{P}_m = \mathcal{N}(0, 1)$ . If the means of the two distribution are too far apart the training process becomes unstable, hence it affects the kurtosis and skewness of the imposed distributions.

**Testing** BioMetricNet, in the current setting, has been developed for 1:1 verification in a face authentication scenario, particularly, when there is a single image template per subject. Therefore, BioMetricNet has been validated on 6 popular unconstrained face datasets for 1:1 verification, excluding large scale datasets like MegaFace [25] and IJB [26] used for set-based face recognition, i.e., deciding whether two *sets* of images of a face belong to the same person or not.

Labeled Faces in the Wild (LFW) [27] and YouTube Faces (YTF) [28] are the most commonly used datasets for unconstrained face verification on images and videos. LFW consists of 13233 face images collected from 5749 people. YTF consists of 3425 videos of 1595 people. Latest deep learning models for face verification are powerful enough to achieve almost perfect accuracy on LFW and YTF, making the related results not very informative. For detailed insights we further test BioMetricNet on more challenging datasets such as Cross-Age LFW (CALFW) [29] which is constructed by selecting 3000 positive face pairs with age gap from LFW to add the aging process to intraclass variance, and Cross-Pose LFW (CPLFW) [30] which is constructed from 3000 face pairs of LFW with pose difference to add pose variation to intra-class variance. Finally, we evaluate our method on Celebrity dataset in frontal and profile views (CFP) [31] having 500 identities with 7000 images, and in-the-wild age database (AgeDB) [32] containing 16488 images of 568 identities. For all the datasets, we report the results for 6000 pairs of testing images and videos having 3000 matching and 3000 non-matching pairs. For reporting the performance we follow the standard protocol of *unrestricted with labeled outside data* as done in [5, 9, 10].

### 3.4 Effect of Feature Vector Dimensionality

We explored the effect of different dimensionality of the feature vector by fixing  $p = 1$ , and varying  $d$ , see Tab. 1. It can be observed that small values of  $d$  are not sufficient to capture the most discriminative facial features. On the other

Table 1: Accuracy (%) for different feature vector  $d$  and latent vector  $p$  dimensionality. Highest accuracy is obtained for the feature vector of size  $d = 512$  and for  $p = 1$

| Dataset | $d = 128$ | $d = 256$ | $d = 512$    | $d = 1024$ | $p = 1$      | $p = 3$ | $p = 8$ | $p = 16$     |
|---------|-----------|-----------|--------------|------------|--------------|---------|---------|--------------|
| LFW     | 99.47     | 99.51     | <b>99.80</b> | 99.63      | <b>99.80</b> | 99.75   | 99.74   | 99.72        |
| YTF     | 97.57     | 97.76     | <b>98.06</b> | 98.0       | <b>98.06</b> | 97.85   | 97.73   | 97.76        |
| CALFW   | 96.48     | 96.59     | <b>97.07</b> | 96.78      | <b>97.07</b> | 97.02   | 96.92   | 96.93        |
| CPLFW   | 94.89     | 94.81     | <b>95.60</b> | 95.25      | <b>95.60</b> | 95.57   | 95.13   | 95.43        |
| CFP-FP  | 99.01     | 99.08     | <b>99.35</b> | 99.25      | 99.35        | 99.33   | 99.33   | <b>99.47</b> |

hand, a too large feature space (1024) causes overfitting and thus a performance drop. We picked the best value, i.e.  $d = 512$ , since in our experiments this choice leads to the highest accuracy.

### 3.5 Effect of Latent Space Dimensionality

In order to select the optimal latent space size we explored different dimensionalities by fixing  $d = 512$ . The results are shown in Tab. 1. In this case, as a general behavior it can be observed that an increase in  $p$  leads to a performance drop.

Since  $p$  affects the number of parameters at the very bottom of MetricNet (an FC network), its choice strongly affects the overall performance. We conjecture that large values of  $p$  might be beneficial for very complex dataset, for which the amount of training data is typically large. Indeed, samples in a higher dimensional latent space are generally more linearly separable. This is even more important when the number of data points is very large. On the other hand, too large values of  $p$  might lead to a performance drop as it becomes difficult to learn a mapping onto large latent space. From Tab. 1 it can be seen that for most of the datasets,  $p = 1$  is sufficient. On the other hand, for CFP-FP it can be seen that the highest accuracy (even though by a small amount) is reached for  $p = 16$ . In this case, a higher latent space dimensionality provides room to achieve a better separation. Since  $p = 1$  provides optimal or close to optimal results in all cases, we choose this value for the experiments.

### 3.6 Parameters of Target Distributions

In this section, we perform an experiment to explore the behavior of different parameters of the target distributions. At first, let us recall that we set the two distributions to have the same variance  $\sigma_m = \sigma_n = \sigma$ . This allows us to have only a single free parameter, i.e. the ratio  $(\mu_m - \mu_n)/\sigma$ , affecting our design, in terms of how far apart we place the distributions compared to the chosen variance. Without loss of generality, this can be tested by setting the distributions to be  $\mathbb{P}_m = \mathcal{N}(0, 1)$  and  $\mathbb{P}_n = \mathcal{N}(w, 1)$ , where  $w = [0.5, 120]$ . From now on if not differently specified we will consider  $p = 1$  and  $d = 512$ .

Table 2: **Verification accuracy %** of different methods on LFW, YTF, CALFW, CPLFW, CFP-FP and AgeDB. BioMetricNet achieves state-of-the-art results for YTF, CALFW, CPLFW, CFP-FP, and AgeDB and obtains similar accuracy to the state-of-the-art for LFW

| Method            | # Image | LFW          | YTF          | CALFW        | CPLFW        | CFP-FP       | AgeDB        |
|-------------------|---------|--------------|--------------|--------------|--------------|--------------|--------------|
| SphereFace [8]    | 0.5M    | 99.42        | 95.0         | 90.30        | 81.40        | 94.38        | 91.70        |
| SphereFace+ [33]  | 0.5M    | 99.47        | -            | -            | -            | -            | -            |
| FaceNet [5]       | 200M    | 99.63        | 95.10        | -            | -            | -            | 89.98        |
| VGGFace [1]       | 2.6M    | 98.95        | 97.30        | 90.57        | 84.00        | -            | -            |
| DeepID [2]        | 0.2M    | 99.47        | 93.20        | -            | -            | -            | -            |
| ArcFace [9]       | 5.8M    | <b>99.82</b> | 98.02        | 95.45        | 92.08        | 98.37        | 95.15        |
| CenterLoss [34]   | 0.7M    | 99.28        | 94.9         | 85.48        | 77.48        | -            | -            |
| DeepFace [35]     | 4.4M    | 97.35        | 91.4         | -            | -            | -            | -            |
| Baidu [36]        | 1.3M    | 99.13        | -            | -            | -            | -            | -            |
| RangeLoss [37]    | 5M      | 99.52        | 93.7         | -            | -            | -            | -            |
| MarginalLoss [38] | 3.8M    | 99.48        | 95.98        | -            | -            | -            | -            |
| CosFace [10]      | 5M      | 99.73        | 97.6         | -            | -            | 95.44        | -            |
| BioMetricNet      | 3.8M    | <b>99.80</b> | <b>98.06</b> | <b>97.07</b> | <b>95.60</b> | <b>99.35</b> | <b>96.12</b> |

In more detail, in Fig. 5 we show the skewness and kurtosis of the latent representation as a function of  $w$  for LFW dataset. It can be seen that in the region corresponding to  $20 \leq w \leq 90$  the skewness and kurtosis are close to 0 and 3, respectively, and the accuracy is high, showing that the training indeed converges to Gaussian distributions. We eventually choose  $\mu_m = 0$  and  $\mu_n = 40$  to keep the distributions sufficiently far apart from each other. Further, if the difference between  $\mu_m$  and  $\mu_n$  is too large (e.g.  $\mu_n > 90$ ), the training process becomes unstable and the distributions become far from Gaussian.

### 3.7 Performance Comparison

Tab. 2 reports the maximum verification accuracy obtained for different methods on several datasets. For YTF and LFW as reported in Tab. 2, it can be observed that BioMetricNet achieves higher accuracy with respect to other methods. In particular, it achieves an accuracy of 98.06% and 99.80% for YTF and LFW datasets respectively. On these two datasets, ArcFace obtains a comparable accuracy.

For a more in-depth comparison we further test BioMetricNet on more challenging datasets, i.e. CALFW, CPLFW, CFP-FP and AgeDB. State-of-the-art results on these datasets are far from the “almost perfect” accuracy we previously observed. In Tab. 2 we compare the verification performance for these datasets. As can be observed, BioMetricNet significantly outperforms the baseline methods (CosFace, ArcFace, and SphereFace). For CPLFW, BioMetricNet achieves an accuracy of 95.60% obtaining an error rate that is 3.52% lower than previous state-of-the-art results, outperforming ArcFace by a significant margin. For CALFW, BioMetricNet achieves an accuracy of 97.07% which is 1.62% lower than previous state-of-the-art results. For CFP dataset BioMetricNet achieves an accuracy of 99.35% lowering the error rate by about 1% with respect to ArcFace. Finally, for AgeDB BioMetricNet achieves an accuracy of 96.12% lowering the error rate by about 1% compared to ArcFace.

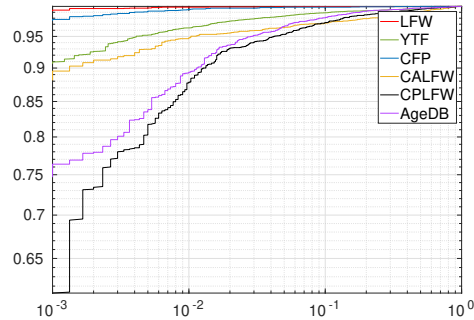


Fig. 6: ROC curve of BioMetricNet on LFW, YTF, CFP, CALFW, CPLFW and AgeDB.

Table 3: GAR obtained for LFW, YTF, CFP, CALFW, CPLFW and AgeDB at  $\text{FAR}=\{10^{-2}, 10^{-3}\}$

| Dataset  | GAR@ $10^{-2}$ FAR% | GAR@ $10^{-3}$ FAR% |
|----------|---------------------|---------------------|
| LFW      | 99.87               | 99.20               |
| YTF      | 96.93               | 90.87               |
| CALFW    | 94.63               | 88.13               |
| CPLFW    | 87.73               | 61.27               |
| CFP-FP   | 99.43               | 97.57               |
| AgeDB-30 | 89.23               | 74.70               |

To summarise, when compared to state-of-art approaches BioMetricNet consistently achieves higher accuracy, proving that, by learning the metric to be used to compare facial features in a regularized space, the discrimination ability of the network is increased. This becomes more evident on more challenging datasets where the gap from perfect accuracy is larger.

### 3.8 ROC Analysis

The Receiver Operating Characteristic (ROC) analysis of BioMetricNet is illustrated in Fig. 6. This curve depicts the Genuine Acceptance Rate (GAR), namely the relative number of correctly accepted matching pairs as function of the False Acceptance Rate (FAR), the relative number of incorrectly accepted non-matching pairs. Furthermore, we report the GAR at different FAR values, namely  $\text{FAR}=\{10^{-2}, 10^{-3}\}$  in Tab. 3. By means of the ROC, we can analyze how the verification task solved by BioMetricNet generalizes across different datasets. It is immediate to notice that, as a result of clear separation and low contamination of the area between the matching and non-matching distributions, high GARs are obtained at low FARs. This is generally true at different “complexity” levels as exposed by the considered datasets. More in detail, for LFW at  $\text{FAR}=10^{-2}$  and  $\text{FAR}=10^{-3}$  high GARs of 99.87% and 99.20% are obtained, see Tab. 3. For YTF at  $\text{FAR}=10^{-2}$  and  $\text{FAR}=10^{-3}$ , GARs of 96.93% and

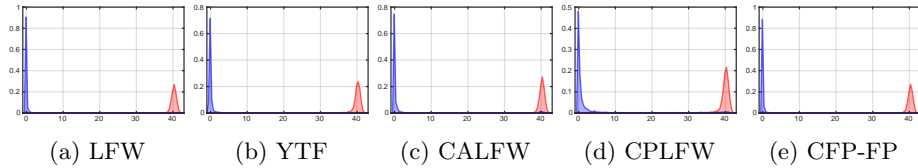


Fig. 7: Histogram of  $\mathbf{z}$  decision statistics of BioMetricNet matching and non-matching pairs from (a) LFW; (b) YTF; (c) CALFW; (d) CPLFW; (e) CFP-FP. Blue area indicates matching pairs while red indicates non-matching pairs.

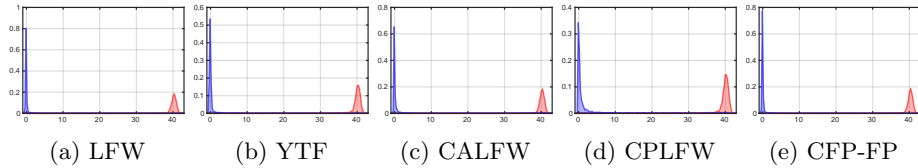


Fig. 8: Histogram of  $\bar{\mathbf{z}}$  decision statistics of BioMetricNet matching and non-matching pairs from (a) LFW; (b) YTF; (c) CALFW; (d) CPLFW; (e) CFP-FP. Blue area indicates matching pairs while red indicates non-matching pairs.

90.87% are obtained. The same behavior can be observed for CFP. For challenging datasets of CALFW CPLFW and AgeDB it can be observed that the ROC curves obtained are comparatively lower compared to LFW, YTF, and CFP. The GARs at  $\text{FAR}=10^{-2}$  and  $\text{FAR}=10^{-3}$  comes to be 94.63% and 88.13% for CALFW, 87.73% and 61.27% for CPLFW and 89.23% and 74.70% for AgeDB respectively.

### 3.9 Analysis of Metrics Distribution

BioMetricNet closely maps the metrics for matching and non-matching pairs onto the imposed target Gaussian distributions. To analyze more in depth the effects of the latent space regularization, we depict the histograms of  $\mathbf{z}$  and  $\bar{\mathbf{z}}$  computed over different test datasets, in Fig. 7 and Fig. 8 respectively. At first, it can be noticed that for both  $\mathbf{z}$  and  $\bar{\mathbf{z}}$  the proposed regularisation is able to shape the latent space as intended by providing Gaussian-shaped distributions. Observing the histograms of  $\mathbf{z}$  and  $\bar{\mathbf{z}}$ , it can be noticed that for all the datasets BioMetricNet very effectively separates matching and non-matching pairs.

Concerning non-matching pairs, the distributions of  $\mathbf{z}$  are indeed Gaussian with the chosen parameters. For matching pairs, it can be observed that the  $\mathbf{z}$  score has the correct mean, but tends to have a lower variance than the target distribution. A possible explanation is that matching and non-matching pairs exhibit different variability, so it is difficult to match them to distributions with the same variance. Indeed, for a fixed number of persons, the number of possible non-matching pairs is much larger than the number of possible matching pairs. Moreover, the KL divergence is not symmetric and the chosen loss tends

to promote sample distributions with a smaller variance than the target one, rather than with a larger variance. Hence, a solution where matching pairs have a smaller variance than the target distribution is preferred with respect to a solution where non-matching pairs have a larger variance than the target distribution. We can also observe that for more difficult datasets, like CALFW and CPLFW, the distribution obtained for matching pairs has heavier tails than the target distribution.

The histogram for  $\bar{\mathbf{z}}$  scores in Fig. 8 shows that the variance of both matching and non-matching pairs is slightly reduced with respect to that of  $\mathbf{z}$ . Since reduced variance means increased verification accuracy, this justifies using  $\bar{\mathbf{z}}$  over  $\mathbf{z}$ . Furthermore, the decision boundary we are using depends only on mean values, which are preserved, and thus it is not affected by the slight decrease in variance.

## 4 Conclusions

We have presented a novel and innovative approach for unconstrained face verification mapping learned discriminative facial features onto a regularized metric space, in which matching and non-matching pairs follow specific and well-behaved distributions. The proposed solution, which does not impose a specific metric, but allows the network to learn the best metric given the target distributions, leads to improved accuracy compared to the state of the art. In BioMetricNet distances between input pairs behave more regularly, and instead of learning a complex partition of the input space, we learn a complex metric over it which further enables the use of much simpler boundaries in the decision phase. With extensive experiments, on multiple datasets with several state-of-the-art benchmark methods, we showed that BioMetricNet consistently outperforms other existing techniques. Future work will consider BioMetricNet in the context of 3D face verification and adversarial attacks. Moreover, considering the slight mismatch between metric distributions and target distributions, it is worth investigating if alternative parameter choices for the target distributions can lead to improved results.

## 5 Acknowledgment

This work results from the research cooperation with Sony R&D Center Europe Stuttgart Laboratory 1.

## References

1. Parkhi, O.M., Vedaldi, A., Zisserman, A., et al.: Deep Face Recognition. In: British Machine Vision Conference. Volume 1. (2015) 6
2. Sun, Y., Chen, Y., Wang, X., Tang, X.: Deep Learning Face Representation by Joint Identification-Verification. In: Advances in Neural Information Processing Systems. (2014) 1988–1996
3. Sun, Y., Liang, D., Wang, X., Tang, X.: DeepID3: Face Recognition with very Deep Neural Networks. arXiv preprint arXiv:1502.00873 (2015)
4. Sun, Y., Wang, X., Tang, X.: Deeply learned face representations are sparse, selective, and robust. In: Proceedings of the IEEE conference on Computer Vision and Pattern Recognition. (2015) 2892–2900
5. Schroff, F., Kalenichenko, D., Philbin, J.: FaceNet: A Unified Embedding for Face Recognition and Clustering. In: Proceedings of the IEEE conference on Computer Vision and Pattern Recognition. (2015) 815–823
6. Wang, J., Zhou, F., Wen, S., Liu, X., Lin, Y.: Deep Metric Learning with Angular Loss. In: Proceedings of the IEEE International Conference on Computer Vision. (2017) 2593–2601
7. Liu, W., Wen, Y., Yu, Z., Yang, M.: Large-Margin Softmax Loss for Convolutional Neural Networks. In: International Conference on Machine Learning. Volume 2. (2016) 7
8. Liu, W., Wen, Y., Yu, Z., Li, M., Raj, B., Song, L.: SpheroFace: Deep Hypersphere Embedding for Face Recognition. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. (2017) 212–220
9. Deng, J., Guo, J., Xue, N., Zafeiriou, S.: ArcFace: Additive Angular Margin Loss for Deep Face Recognition. In: Proceedings of the IEEE conference on Computer Vision and Pattern Recognition. (2019) 4690–4699
10. Wang, H., Wang, Y., Zhou, Z., Ji, X., Gong, D., Zhou, J., Li, Z., Liu, W.: CosFace: Large Margin Cosine Loss for Deep Face Recognition. In: Proceedings of the IEEE conference on Computer Vision and Pattern Recognition. (2018) 5265–5274
11. Ustinova, E., Lempitsky, V.: Learning deep embeddings with histogram loss. In: Advances in Neural Information Processing Systems. (2016) 4170–4178
12. Testa, M., Ali, A., Bianchi, T., Magli, E.: Learning mappings onto regularized latent spaces for biometric authentication. In: Proceedings of the IEEE International Workshop on Multimedia Signal Processing. (2019)
13. Ali, A., Testa, M., Bianchi, T., Magli, E.: Authnet: Biometric authentication through adversarial learning. In: 2019 IEEE 29th International Workshop on Machine Learning for Signal Processing (MLSP), IEEE (2019) 1–6
14. Neal, R.M.: Bayesian Learning for Neural Networks. Volume 118. Springer Science & Business Media (2012)
15. Szegedy, C., Ioffe, S., Vanhoucke, V., Alemi, A.A.: Inception-v4, Inception-Resnet and the Impact of Residual Connections on Learning. In: Thirty-First AAAI Conference on Artificial Intelligence. (2017)
16. Chen, K., Tao, W.: Once for all: a two-flow convolutional neural network for visual tracking. *IEEE Transactions on Circuits and Systems for Video Technology* **28**(12) (2017) 3377–3386
17. Held, D., Thrun, S., Savarese, S.: Learning to track at 100 fps with deep regression networks. In: European Conference on Computer Vision, Springer (2016) 749–765
18. Kingma, D.P., Ba, J.: Adam: A Method for Stochastic Optimization. arXiv preprint arXiv:1412.6980 (2014)

19. LeCun, Y., Boser, B., Denker, J.S., Henderson, D., Howard, R.E., Hubbard, W., Jackel, L.D.: Backpropagation Applied to Handwritten Zip Code Recognition. *Neural Computation* **1**(4) (1989) 541–551
20. Rumelhart, D.E., Hinton, G.E., Williams, R.J., et al.: Learning representations by back-propagating errors. *Cognitive modeling* **5**(3) (1988) 1
21. Abadi, M., Barham, P., Chen, J., Chen, Z., Davis, A., Dean, J., Devin, M., Ghemawat, S., Irving, G., Isard, M., et al.: TensorFlow: A System for Large-Scale Machine Learning. In: 12th {USENIX} Symposium on Operating Systems Design and Implementation. (2016) 265–283
22. Zhang, K., Zhang, Z., Li, Z., Qiao, Y.: Joint Face Detection and Alignment using Multitask Cascaded Convolutional Networks. *IEEE Signal Processing Letters* **23**(10) (2016) 1499–1503
23. Yi, D., Lei, Z., Liao, S., Li, S.Z.: Learning Face Representation from Scratch. arXiv preprint arXiv:1411.7923 (2014)
24. <http://trillionpairs.deepglint.com/overview>
25. Kemelmacher-Shlizerman, I., Seitz, S.M., Miller, D., Brossard, E.: The megaface benchmark: 1 million faces for recognition at scale. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. (2016) 4873–4882
26. Maze, B., Adams, J., Duncan, J.A., Kalka, N., Miller, T., Otto, C., Jain, A.K., Niggel, W.T., Anderson, J., Cheney, J., et al.: Iarpa janus benchmark-c: Face dataset and protocol. In: 2018 International Conference on Biometrics (ICB), IEEE (2018) 158–165
27. Huang, G.B., Mattar, M., Berg, T., Learned-Miller, E.: Labeled Faces in the Wild: A Database for Studying Face Recognition in Unconstrained Environments. (2008)
28. Wolf, L., Hassner, T., Maoz, I.: Face Recognition in Unconstrained Videos with Matched Background Similarity. *IEEE* (2011)
29. Zheng, T., Deng, W., Hu, J.: Cross-Age LFW: A Database for Studying Cross-Age Face Recognition in Unconstrained Environments. arXiv preprint arXiv:1708.08197 (2017)
30. Zheng, T., Deng, W.: Cross-Pose LFW: A Database for Studying Crosspose Face Recognition in Unconstrained Environments. Beijing University of Posts and Telecommunications, Tech. Rep (2018) 18–01
31. Sengupta, S., Chen, J.C., Castillo, C., Patel, V.M., Chellappa, R., Jacobs, D.W.: Frontal to Profile Face Verification in the Wild. In: 2016 IEEE Winter Conference on Applications of Computer Vision, IEEE (2016) 1–9
32. Moschoglou, S., Papaioannou, A., Sagonas, C., Deng, J., Kotsia, I., Zafeiriou, S.: Agedb: the first manually collected, in-the-wild age database. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops. (2017) 51–59
33. Liu, W., Lin, R., Liu, Z., Liu, L., Yu, Z., Dai, B., Song, L.: Learning Towards Minimum Hyperspherical Energy. In: Advances in Neural Information Processing Systems. (2018) 6222–6233
34. Wen, Y., Zhang, K., Li, Z., Qiao, Y.: A Discriminative Feature Learning Approach for Deep Face Recognition. In: European Conference on Computer Vision, Springer (2016) 499–515
35. Taigman, Y., Yang, M., Ranzato, M., Wolf, L.: DeepFace: Closing the Gap to Human-Level Performance in Face Verification. In: Proceedings of the IEEE conference on Computer Vision and Pattern Recognition. (2014) 1701–1708
36. Liu, J., Deng, Y., Bai, T., Wei, Z., Huang, C.: Targeting Ultimate Accuracy: Face Recognition via Deep Embedding. arXiv preprint arXiv:1506.07310 (2015)

37. Zhang, X., Fang, Z., Wen, Y., Li, Z., Qiao, Y.: Range Loss for Deep Face Recognition with Long-tailed Training Data. In: Proceedings of the IEEE International Conference on Computer Vision. (2017) 5409–5418
38. Deng, J., Zhou, Y., Zafeiriou, S.: Marginal Loss for Deep Face Recognition. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops. (2017) 60–68