

von Mises-Fisher Mixture Model-based Deep learning: Application to Face Verification

Md. Abul Hasnat, Julien Bohné, Jonathan Milgram, Stéphane Gentric and Liming Chen

arXiv:1706.04264v2 [cs.CV] 31 Dec 2017

Abstract—A number of pattern recognition tasks, *e.g.*, face verification, can be boiled down to classification or clustering of unit length directional feature vectors whose distance can be simply computed by their angle. In this paper, we propose the von Mises-Fisher (vMF) mixture model as the theoretical foundation for an effective deep-learning of such directional features and derive a novel vMF Mixture Loss and its corresponding vMF deep features. The proposed vMF feature learning achieves the characteristics of discriminative learning, *i.e.*, compacting the instances of the same class while increasing the distance of instances from different classes. Moreover, it subsumes a number of popular loss functions as well as an effective method in deep learning, namely normalization. We conduct extensive experiments on face verification using 4 different challenging face datasets, *i.e.*, LFW, YouTube faces, CACD and IJB-A. Results show the effectiveness and excellent generalization ability of the proposed approach as it achieves state-of-the-art results on the LFW, YouTube faces and CACD datasets and competitive results on the IJB-A dataset.

Index Terms—Deep Learning, Face Recognition, Mixture Model, von Mises-Fisher distribution.

1 INTRODUCTION

A number of pattern recognition tasks, *e.g.*, face recognition (FR), can be boiled down to supervised classification or unsupervised clustering of unit length feature vectors whose distance can be simply computed by their angle, *i.e.*, cosine distance. In deep learning based FR, numerous methods find it useful to unit-normalize the final feature vectors, *e.g.*, [14], [38], [48], [65], [66], [68], [73]. Besides, the widely used and simple softmax loss has been extended with additional or reinforced supervising signal, *e.g.*, center loss [68], large margin softmax loss [39], to further enable a discriminative learning, *i.e.*, compacting intra-class instances while repulsing inter-class instances, and thereby increase the final recognition accuracy. However, the great success of these methods and practices remains unclear from a theoretical viewpoint, which motivates us to *study the deep feature representation from a theoretical perspective*.

Statistical Mixture Models (MM) is a common method to perform probabilistic clustering and widely used in data mining and machine learning [44]. MM plays key role in model based clustering [6], which assumes a generative

model, *i.e.*, each observation is a sample from a finite mixture of probability distributions. In this paper, we adopt the *theoretical concept of MM to model the deep feature representation task* and realize the relationship among the model parameters and deep features.

Unit length normalized feature vectors are directional features which only keep the orientations of data features as discriminative information while ignoring their magnitude. In this case, simple angle measurement, *e.g.*, cosine distance, can be used as dissimilarity measure of two data points and provides very intuitive geometric interpretation of similarity [17]. In this paper, we propose to model the (deep)-features delivered by the deep neural nets, *e.g.*, CNN-based neural networks, as a mixture of von Mises-Fisher distributions, also called vMF Mixture Model (vMFMM). The von Mises-Fisher (vMF) is a fundamental probability distribution, which has been successfully used in numerous unsupervised classification tasks [3], [17], [21]. By combining this vMFMM with deep neural networks, we derive a novel loss function, namely vMF mixture loss (vMFML) which enables a discriminative learning. Figure 1(a) (from right to left) provides an illustration of the proposed model. Figure 1(b) shows¹ the discriminative nature of the proposed model, *i.e.*, the learned features for each class are compacted whereas inter-class features are repulsed.

To demonstrate the effectiveness of the proposed method, we carried out extensive experiments on face recognition (FR) task on which recent deep learning-based methods [36], [45], [54], [55], [58] have surpassed human level performance. We used 4 different challenging face datasets, namely LFW [25] for single image based face verification, IJB-A [30] for face templates matching, YouTube Faces [69] (YTF) for video faces matching, and CACD [7] for cross age face matching. Using only one deep CNN model trained on the MS-Celeb dataset [19], the proposed method achieves 99.63% accuracy on LFW, 85% TAR@FAR=0.001 on IJB-A [30], 96.46% accuracy on YTF [69] and 99.2% accuracy on CACD [7]. These results indicate that our method generalizes very well across different datasets as it achieves state-of-the-art results on the LFW, YouTube faces and CACD datasets and competitive results on the IJB-A dataset.

Laboratoire LIRIS, École centrale de Lyon, 69134 Ecully, France.
Safran Identity & Security, 92130 Issy-les-Moulineaux, France.
e-mail:md-abul.hasnat@ec-lyon.fr, julien.bohne@safrangroup.com,
jonathan.milgram@safrangroup.com, stephane.gentric@safrangroup.com
liming.chen@ec-lyon.fr

1. We visualize the features of the MNIST digits in the 3D space (similar illustration with 2D plot in [68]). The CNN is composed of 6 convolution, 2 pool and 1 FC (with 2 neurons for 2D visualization) layers. We optimize it using the softmax loss and the proposed vMFML.

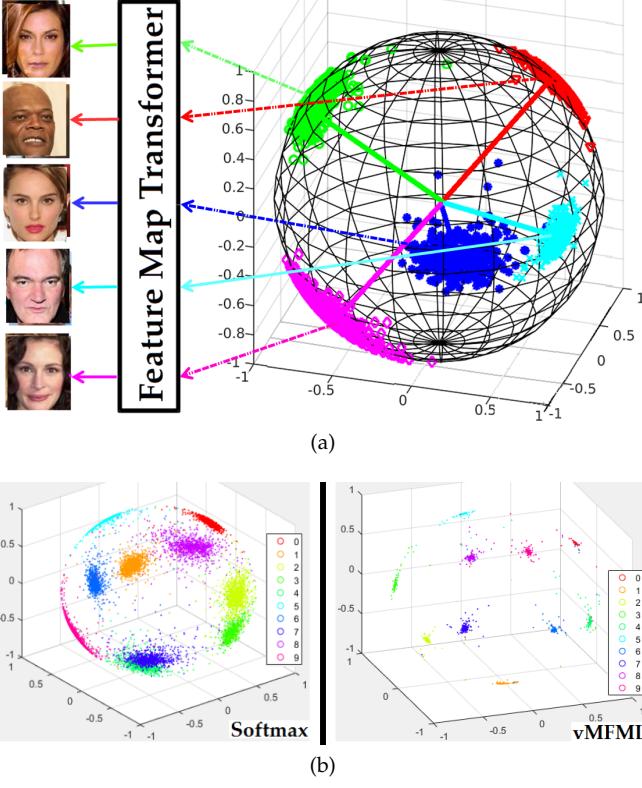


Fig. 1: (a) illustration of the proposed model with a 5 classes vMFMM, where *facial identity* represents the *classes* and (b) illustration of the 3D features learned from the MNIST digits [33], left: softmax loss, right: proposed (vMFML). Features from different classes are shown with different markers and colors in the sphere. See Appendix A for an extended experiment and further illustrations.

The contributions of the proposed method can be summarized as follows:

- we propose a novel *feature representation* model (Sect. 3.1.1) from a theoretical perspective. It comprises the statistical mixture model [44] with directional distribution [41]. It provides a novel view to understand and model the desired pattern classification task. Therefore, it can help to develop efficient methods to achieve better results.
- we propose a *directional feature representation learning* method, called vMF-FL (Sect. 3.1.2), which combines the theoretical model with the CNN model. vMF-FL provides a novel loss function, called vMFML (Sect. 3.2), whose formulation *w.r.t.* the backpropagation [33] method shows that it can be easily integrated with any CNN model. Moreover, vMFML is able to explain (Sect. 3.3) different loss functions [38], [39], [48], [66], [68] and normalization methods [12], [28], [51]. vMFML not only interprets the relation among the parameters and features, but also improves the CNN learning task *w.r.t.* efficiency (faster convergence) and performance (better accuracy, Sect. 4.2.1). Therefore, it can be used in a variety of classification tasks under the assumption of directional features.
- we verify (Sect. 4.2) the effectiveness of the proposed

vMFML on the task of face verification through comprehensive experiments on four face benchmarks depicting various challenges, *e.g.*, pose, lighting, age, and demonstrate its generalization skills across datasets.

- we perform (sect.4.3) additional experiments and provide in-depth analysis of various factors, *e.g.*, sensitivity of parameters, training data size, failure cases, to further gain insight of the proposed method.

In the remaining part, we study the related work in Section 2, describe our method in Section 3, present experimental results with analysis in Section 4 and finally draw conclusions in Section 5.

2 RELATED WORK

2.1 Mixture models and Loss Functions

Mixture models (MM) [44] have been widely used in numerous machine learning problems, such as classification [15], clustering [6], image analysis [21], text analysis [3], shape retrieval [37], etc. However, their potentials are relatively under-explored with the neural network (NN) based learning methods. Several recent work [46], [60], [62], [63] explored the concept of MM with NN from different perspectives. [62] used the Gaussian MM (GMM) to model deep NN as a mixture of transformers. [46] aimed to capture the variations in the nuisance variables (*e.g.*, object poses) and used NN as a rendering function to propose deep rendering MM. Both of these methods use MM within the NN, whereas *we consider NN as a single transformer*. [60], [63] used GMM with NN and applied it for speech analysis. [63] performed discriminative feature learning task via the proposed GMM layer. [60] used the concept of log-linear model with GMM and modified the softmax loss accordingly. While our method is more similar to [60], [63], there are several differences: (a) we use directional (unit normalized) features; (b) we use the vMF [41] distribution which is more appropriate for directional features [22]; (c) our feature representation model is based on a generative model-based concept; and (d) we exploit the CNN model and explore practical application of computer vision.

MM with directional distributions [41] have been used in a variety of domains to analyze images [22], speech [64], text [3], [17], gene expressions [3], shapes [47], pose [16], diffusion MRI [5], etc. However, they remain unexplored to learn discriminative features. In this paper, we aim to explore it by modeling the feature learning task with the vMF distribution [3], [41] and combining it with a CNN model. To the best of our knowledge, *this is the first reported attempt to use the directional distribution with the CNN model*.

Loss functions are essential part of CNN training. The CNN model parameters are learned by optimizing certain loss functions which are defined based on the given task (*e.g.*, classification, regression) and the available supervisory information (*e.g.*, class labels, price). While the softmax loss [18] is most commonly incorporated, recent researches [39], [68] indicate that it cannot guarantee to provide discriminative features. [68] proposed center loss as a supplementary loss to minimize intra-class variations and hence to improve feature discrimination. Our method *performs the same task*

with the concentration (κ) parameter and requires no supplementary loss function, see Section 3.3 for details. [38], [39] proposed the large-margin softmax loss by incorporating an intuitive margin on the classification boundary, which can be explained by our method under certain condition, see Section 3.3 for details.

2.2 Face Recognition

Face recognition (FR) is a widely studied problem in which remarkable results have been achieved by the recent deep CNN based methods on several standard FR benchmarks datasets, such as, the Labeled Faces in the Wild (LFW) [25], YouTube Faces (YTF) [69] and IARPA Janus Benchmark A (IJB-A) [30]. In Table 1, we study² these methods and decompose their key aspects as: (a) CNN model design; (b) objective/loss functions; (c) fine-tune and additional learning method; (d) multi-modal input and number of CNNs and (e) use of the training database.

Recent deep CNN based FR methods tend to adopt (directly or slightly modify) the famous CNN architectures [18] proposed for the ImageNet [50] challenge. The CNN Info column of Table 1 provides the details of CNN models used by different FR methods. The AlexNet model is used by [1], [40], [42], [49], [52], [53], [54], VGGNet model is used by [1], [13], [14], [42], [43], [45], [57], GoogleNet model is used by [54], [71] and ResNet model is used by [20], [38], [48], [66], [68], [76]. Besides the famous CNN models, [73] proposed a simpler CNN model which is used by [9], [14], [65]. CNNs with lower depth have been used by [55], [56], [58], [59], [68], where the model complexity is increased with locally connected convolutional layers. Parallel CNNs have been employed by [36], [78] to simultaneously learn features from different facial regions. *We adopt the ResNet [24] based deeper CNN model.*

FR methods are trained with different loss functions, see *Loss function* column in Table 1. Most FR methods learn the facial feature representation model by training the CNN for identity classification. For this purpose, the softmax loss [18] is used to optimize the classification objective, which requires the facial images with associated identity labels. Several variants of the softmax loss [38], [39], [48], [66] have been recently proposed to enhance its discriminability. Besides, the contrastive loss [11], [18] is used by [55], [56], [57], [58], [71] for face verification, which requires similar/dissimilar face image pairs and similarity labels. Moreover, the triplet loss [54] is used by [13], [36], [45], [54] for face verification and requires the face triplets (*i.e.* anchor, positive, negative). Recently, [66] proposed different formulations of the contrastive and triplet losses such that they can be trained with only the identity labels. *Our proposed vMFML, simply learns the features via identity classification and requires only the class labels.* Moreover, the theoretical foundation of vMFML provides interesting interpretation and relationship with the very recently proposed softmax based loss functions, such as [38], [39], [48], [66], see Sect. 3.3 for details.

CNN training with multiple loss functions have been adopted by several methods, where the losses are optimized

2. We only consider the deep CNN based methods. For the others, we refer readers to the survey [32] for LFW and the [30] for IJB-A.

jointly [49], [55], [56], [57], [68], [76] or sequentially (optimize softmax loss followed by the other loss) [13], [36], [45], [58], [71]. However, multiple loss optimization not only requires additional efforts for training data preparation but also complicates the CNN training procedure [20]. *Our FR method optimizes single loss and do not need extra efforts.*

After training the CNN model, a second learning method is often incorporated by different FR methods, see *Additional Learning* column of Table 1. CNN fine-tune is a particular form of transfer learning commonly employed by several methods [9], [52], [65]. It consists of updating the trained CNN parameters using the target specific training data. In order to adapt the CNN features to particular FR tasks, several methods apply additional transformation on the CNN features based on the *metric/distance learning* strategy. The Joint Bayesian method [8] is a popular metric learning method and used by [9], [14], [55], [56], [57], [65], [73]. Specific embedding learning methods [52], [53] have been proposed to learn from face triplets. The template adaptation [13] is another strategy which incorporates an SVM classifier and extends the FR tasks for videos and templates [30]. A different approach learns an aggregation module [71] to compute the similarity scores among two sets of video frames. The principal component analysis (PCA) technique is commonly used [1], [42], [43] to learn a dataset specific projection matrix. The above methods often need to prepare its training data from the target specific datasets. Moreover, they [52], [53] may need to carefully prepare the training data, *e.g.*, triplets. *Our FR method do not incorporate any such additional learning strategies.*

FR methods often accumulate features from a set of independently trained CNN models to construct rich facial descriptors, see # of CNNs column of Table 1. These CNN models are trained with different types of inputs: (a) image-crops based on different facial regions (eyes, nose, lips, *etc.*) [14], [55], [56], [57], [65]; (b) different images modalities, such as 2D, 3D, frontalized and synthesized faces at different poses [1], [14], [42], [58] and (c) different training databases with varying number of images [36], [59]. *Our FR method do not apply these approaches and train single CNN model.*

A large facial image database is significantly important to achieve high accuracy on FR [20], [54], [78]. *Dataset Info* column of Table 1 provides the information of the training datasets used by different FR methods. Currently, several FR datasets [19], [73] are publicly available. Among them, the CASIA-WebFace [73] has been widely used by the recent methods [1], [9], [14], [40], [42], [43], [49], [52], [53], [65], [68], [70], [73] to train CNN with a medium sized database. Besides, the recently released MSCELEB [19] dataset, which provides the largest collection of facial images and identities, becomes the standard choice for large scale training. *We exploit it for developing our FR method.*

3 METHODOLOGY

In this section, first we present the *statistical feature representation* model and then discuss the *facial feature representation learning* method *w.r.t.* the model. Finally, we present the complete face recognition pipeline.

TABLE 1: Overview of the state-of-the-art FR methods. In the second column (**CNN Info**), the shorthand notations mean- C: convolutional layer, FC: fully connected layer, LC: locally connected layer and L: loss layer. In the third column (**Loss Function**), the shorthand notations mean- TL: triplet loss, SL: softmax loss, CL: contrastive loss, CCL: C-contrastive loss, CeL: center loss, ASL: angular softmax loss and RL: range loss. In the fourth column (**Additional Learning**), the shorthand notations mean- JB: joint bayesian, TSE: triplet similarity embedding, TPE: triplet probability embedding, TA: template adaptation and X: no metric learning. We list the methods in a decreasing order based on the number of convolution and FC layers in the CNN model.

FR system	CNN Info	Loss function	Additional learning	# of CNNs	Dataset Info
DeepID2+ [56]	4-C, 1-FC, 2L	SL, CL	JB	25	0.29M, 12K
Deepface [58]	2-C, 3-LC, 1-FC, 1-L	SL, CL	X	3	4.4M, 4K
Webscale [59]	2-C, 3-LC, 2-FC, 1-L	SL	X	4	4.5M, 55K
Center Loss [68]	3-C, 3-LC, 1-FC, 2-L	SL, CeL	X	1	0.7M, 17.2K
FV-TSE [52]	6-C, 2-FC, 1-L	SL	TSE	1	0.49M, 10.5K
FV-TPE [53]	7-C, 2-FC, 1-L	SL	TPE	1	0.49M, 10.5K
VIPLFaceNet [40]	7-C, 2-FC, 1-L	SL	X	1	0.49M, 10.57K
All-In-One [49]	7-C, 2-FC, 8-L	SL	TPE	1	0.49M, 10.5K
CASIA-Webface [73]	10-C, 1-L	SL	JB	1	0.49M, 10.5K
FSS [65]	10-C, 1-L	SL	JB	9	0.49M, 10.57K
Unconstrained FV [9]	10-C, 1-FC, 1-L	SL	JB	1	0.49M, 10.5K
Sparse ConvNet [57]	10-C, 1-FC, 1-L	SL, CL	JB	25	0.29M, 12K
FaceNet [54]	11-C, 3-FC, 1-L	TL	X	1	200M, 8M
DeepID3 [55]	8-C, 4-FC, 2-LC, 2L	SL, CL	JB	25	0.3M, 13.1K
MM-DFR [14]	10-C, 2-FC, 1-L and 12-C, 2-FC, 1-L	SL	JB	8	0.49M, 10.5K
VGG Face [45]	13-C, 3-FC, 2L	SL, TL	X	1	2.6M, 2.6K
FV-TA [13]	13-C, 3-FC, 2L	SL, TL	TA	1	2.6M, 2.6K
MFM-CNN [70]	14-C, 2-FC, 1L	SL	X	1	0.49M, 10.5K
Face-Aug-Pose-Syn [43]	16-C, 3-FC, 1-L	SL	PCA	1	0.49M, 10.57K
Deep Multipose [1]	16-C, 3-FC, 1-L and 5-C, 2-FC, 1-L	SL	PCA	12	0.4M, 10.5K
Pose aware FR [42]	16-C, 3-FC, 1-L	SL	PCA	5	0.49M, 10.5K
Range Loss [76]	27-C, 1-FC, 2-L	SL, RL	X	1	1.5M, 100K
DeepVisage [20]	27-C, 1-FC, 1-L	SL	X	1	4.48M, 62K
NormFace [66]	27-C, 1-FC, 2-L	SL, CCL	X	1	0.49M, 10.5K
Megvii [78]	4×10 , 1-L	SL	X	1	5M, 0.2M
Baidu [36]	4×9 -C, 2-L	SL, TL	X	10	1.2M, 1.8K
NAN [71]	57-C, 5-FC, 3-L	SL, CL	Agg	1	3M, 50K
SphereFace [38]	64-C, 1-FC, 1-L	ASL	X	1	0.49M, 10.5K
L2-Softmax [48]	100-C, 1-FC, 1-L	L2-SL	X	1	0.49M, 10.5K

3.1 Model and Method

3.1.1 Statistical Feature Representation (SFR) Model

We propose the SFR model based on the generative model-based approach [44], where the facial features are issued from a finite statistical mixture of probability distributions. Then, these features are transformed into the 2D image space using a *transformer*. See Appendix A for the experimental proof of concept of the proposed SFR model with MNIST [33] digits experiments.

Figure 1(a) (from right to left) provides an illustration of the SFR model, which considers a mixture of *von Mises-Fisher* (vMF) distributions [41] to model the facial features from different identities/classes. The vMF distribution [41] is parameterized with the mean direction μ (shown as solid lines) and concentration κ (indicates the spread of feature points from the solid line). For the i^{th} facial image features \mathbf{x}_i , called *vMF feature*, we define the SFR model with M classes as:

$$SFR(\mathbf{x}_i | \Theta_M) = \sum_{j=1}^M \pi_j V_d(\mathbf{x}_i | \mu_j, \kappa_j) \quad (1)$$

where π_j , μ_j and κ_j denote respectively the mixing proportion, mean direction and concentration value of the j^{th} class.

Θ_M is the set of model parameters and $V_d(\cdot)$ is the density function of the vMF distribution (see Section 3.2 for details).

The SFR model makes *equal privilege assumption* for the classes, *i.e.*, each class j has *equal appearance* probability π and is distributed with *same concentration* value κ . This assumption is important for *discriminative learning* to make sure that the *supervised* classifier is not biased to any particular class regardless of the number of samples and amount of variations present in the training data for each class. On the other hand, μ_j plays significant role to preserve each identity in its respective sub-space. Therefore, the *generative* SFR model can be used for *discriminative* learning tasks, which can be viewed by reversing the directions in Figure 1(a), *i.e.*, information will flow from left to right. Next we discuss the discriminative features learning task *w.r.t.* the SFR model in details.

3.1.2 vMF Feature Learning (vMF-FL) Method

Figure 2 illustrates the workflow of the vMF-FL method, where the features are learned using an object identity classifier. See Appendix A for an extended view of its relationship with the SFR model. The vMF-FL method consists of two sub-tasks: (1) mapping input 2D object images to *vMF feature* using the CNN model, which we use as the *inverse-transformer* *w.r.t.* the SFR model and (2) classifying features

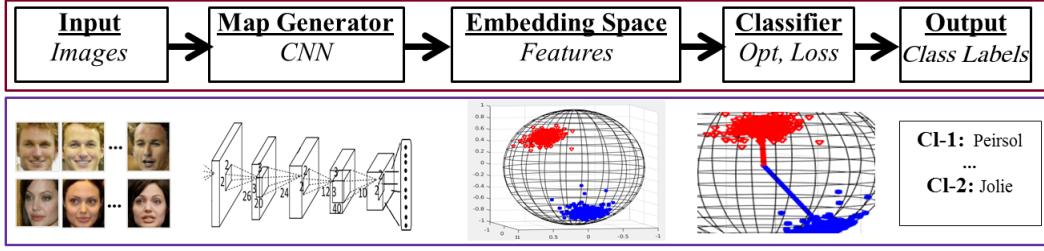


Fig. 2: Workflow of the vMF-FL method. *top*: block diagram and *bottom*: view with an example.

to the respective classes based on the *discriminative* view of the SFR model. It formulates an optimization problem by integrating the SFR and CNN models and learns parameters by minimizing the classification loss. In general, CNN models use the *softmax* function and minimize the cross entropy. Therefore, our integration will replace the *softmax* function according to Eq. 1.

3.1.3 Convolutional Neural Network (CNN) model

The basic ideas of CNN [33] consist of: (a) local receptive fields via convolution and (b) spatial sub-sampling via pooling. At layer l , the convolution of the input $f_{x,y}^{Op,l-1}$ to obtain the k^{th} output map $f_{x,y,k}^{C,l}$ is:

$$f_{x,y,k}^{C,l} = \mathbf{w}_k^l f_{x,y}^{Op,l-1} + b_k^l \quad (2)$$

where, \mathbf{w}_k^l and b_k^l are the shared weights and bias. C denotes convolution and Op (for $l > 1$) denotes various tasks, such as convolution, sub-sampling or activation. For $l = 1$, Op represents the input image. Sub-sampling or pooling performs a simple local operation, *e.g.*, computing the maximum in a local neighborhood followed by reducing spatial resolution as:

$$f_{x,y,k}^{P,l} = \max_{(m,n) \in \mathcal{N}_{x,y}} f_{m,n,k}^{Op,l-1} \quad (3)$$

where, $\mathcal{N}_{x,y}$ is the local neighborhood and P denotes pooling. In order to ensure non-linearity of the network, the feature maps are passed through a non-linear activation function, *e.g.*, the Rectified Linear Unit (ReLU) [18], [23]: $f_{x,y,k}^l = \max(f_{x,y,k}^{l-1}, 0)$.

At the basic level, a CNN is constructed by stacking series of convolution, activation and pooling layers [33]. Often a fully connected (FC) layer is placed at the end, which connects all neurons from the previous layer to all neurons of the next layer.

CNN models are trained by optimizing loss function. The softmax loss, which is widely used for classification, has the following form:

$$\mathcal{L}_{Softmax} = - \sum_{i=1}^N \log \frac{\exp(\mathbf{w}_{y_i}^T f_i + b_{y_i})}{\sum_{l=1}^M \exp(\mathbf{w}_l^T f_i + b_l)} \quad (4)$$

where, f_i and y_i are the features and true class label of the i^{th} image. \mathbf{w}_j and b_j denote the weights and bias of the j^{th} class. N and M denote the number of training samples and the number of classes.

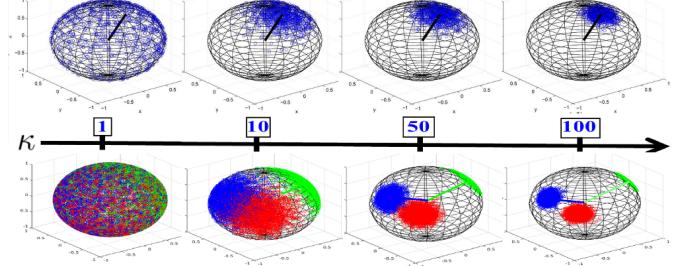


Fig. 3: 3D directional samples from the vMF distribution (above arrow) and the vMFMM of 3 classes (below arrow). Samples are shown on the S^2 sphere for different values of κ .

3.2 SFR model and von Mises-Fisher Mixture Loss (vMFML)

Our proposed SFR model assumes that the facial features are unit vectors and distributed according to a mixture of vMFs. By combining the SFR and CNN, vMF-ML method provides a novel loss function, called the *von Mises-Fisher Mixture Loss* (vMFML). Below we provide its formulation.

3.2.1 vMF Mixture Model (vMFMM)

For a d dimensional random unit vector $\mathbf{x} = [x_1, \dots, x_d]^T \in S^{d-1} \subset \mathbb{R}^d$ (*i.e.*, $\|\mathbf{x}\|_2 = 1$), the density function of the vMF distribution is defined as [41]:

$$V_d(\mathbf{x}|\mu, \kappa) = C_d(\kappa) \exp(\kappa \mu^T \mathbf{x}) \quad (5)$$

where, μ denotes the mean (with $\|\mu\|_2 = 1$) and κ denotes the concentration parameter (with $\kappa \geq 0$). $C_d(\kappa) = \frac{\kappa^{d/2-1}}{(2\pi)^{d/2} I_{d/2-1}(\kappa)}$ is the normalization constant, where, $I_\rho(\cdot)$ is the modified Bessel function of the first kind. The shape of the vMF distribution depends on the value of the concentration parameter κ . For high value of κ , *i.e.* highly concentrated features, the distribution has a mode at the mean direction μ . In contrary, for low values of κ the distribution is uniform, *i.e.* the samples appear as to be uniformly distributed on the sphere. The top row of Figure 3 illustrates examples of the 3D samples in the S^2 sphere, which are distributed according to the vMF distribution with different values of the concentration κ .

Let $\mathbf{X} = \{\mathbf{x}_i\}_{i=1,\dots,N}$ is a set of samples, where N is the total number of samples. For the i^{th} sample \mathbf{x}_i , the vMFMM with M classes is defined³ as [3]: $g_v(\mathbf{x}_i|\Theta_M) = \sum_{j=1}^M \pi_j V_d(\mathbf{x}_i|\mu_j, \kappa_j)$, where $\Theta_M =$

³ Note that, this is equivalent to the definition of SFR model in Eq. 1

$\{(\pi_1, \mu_1, \kappa_1), \dots, (\pi_M, \mu_M, \kappa_M)\}$ is the set of parameters and π_j is the mixing proportion of the j^{th} class. The bottom row of Figure 3 shows the samples from vMFMM with different κ values.

The vMFMM model has been used for *unsupervised classification* [3] to cluster data on the unit hyper-sphere, where the model is estimated by the Expectation Maximization (EM) method. The objective of EM method is to estimate model parameters such that the negative log-likelihood value of the vMFMM model, *i.e.*, $-\log(g_v(\mathbf{X}|\Theta_K))$, is minimized. The EM method estimates the posterior probability in the *E-step* as [3]:

$$p_{ij} = \frac{\pi_j C_d(\kappa_j) \exp(\kappa_j \mu_j^T \mathbf{x}_i)}{\sum_{l=1}^K \pi_l C_d(\kappa_l) \exp(\kappa_l \mu_l^T \mathbf{x}_i)} \quad (6)$$

and model parameters in the *M-step* as [3]:

$$\begin{aligned} \pi_j &= \frac{1}{N} \sum_{i=1}^N p_{ij}, \quad \hat{\mu}_j = \frac{\sum_{i=1}^N p_{ij} \mathbf{x}_i}{\sum_{i=1}^N p_{ij}}, \\ \bar{r} &= \frac{\|\hat{\mu}_j\|}{N \pi_j}, \quad \mu_j = \frac{\hat{\mu}_j}{\|\hat{\mu}_j\|} \text{ and } \kappa_j = \frac{\bar{r} d - \bar{r}^3}{1 - \bar{r}^2} \end{aligned} \quad (7)$$

3.2.2 von Mises-Fisher Mixture Loss (vMFML) and optimization

Our vMF-FL method aims to learn discriminative facial features by minimizing the classification loss. Within this (*supervised classification*) context, we set our objective as to minimize the cross entropy guided by the vMFMM. Therefore, we rewrite the posterior probability based on the *equal privilege assumption* of the SFR model as:

$$p_{ij} = \frac{\exp(\kappa \mu_j^T \mathbf{x}_i)}{\sum_{l=1}^M \exp(\kappa \mu_l^T \mathbf{x}_i)} \quad (8)$$

Now we can exploit the posterior/conditional probability to minimize the cross entropy and define the loss function, called vMFML, as:

$$\begin{aligned} \mathcal{L}_{vMFML} &= - \sum_{i=1}^N \sum_{j=1}^M y_{ij} \log(p_{ij}) \\ &= - \sum_{i=1}^N \log \frac{\exp(\kappa \mu_j^T \mathbf{x}_i)}{\sum_{l=1}^M \exp(\kappa \mu_l^T \mathbf{x}_i)} \\ &= - \sum_{i=1}^N \log \frac{e^{z_{ij}}}{\sum_{l=1}^M e^{z_{il}}} \quad [z_{ij} = \kappa \mu_j^T \mathbf{x}_i] \end{aligned} \quad (9)$$

where, y_{ij} is the true class probability and we set $y_{ij} = 1$ as we only know the true class labels. Now, by comparing the vMFML with the *softmax loss* (Eq. 4), we observe the following differences: (a) vMFML uses unit normalized features: $\mathbf{x} = \frac{\mathbf{f}}{\|\mathbf{f}\|}$; (b) mean parameter has relation with the softmax weight as: $\mu = \frac{\mathbf{w}}{\|\mathbf{w}\|}$; (c) it has no bias and (d) it has an additional parameter κ .

Now, we observe that the proposed vMF-FL method modifies the CNN training by replacing the *softmax loss* with the *vMFML*. Therefore, to learn the parameters we can follow the standard CNN model learning procedure, *i.e.*, iteratively learn through the forward and backward propagation [33]. This requires us to compute the gradients of vMFML *w.r.t.* the parameters. By following the chain rule,

we can compute the gradients (we consider single sample and drop subscript i for brevity) as:

$$\frac{\partial \mathcal{L}}{\partial p_j} = -\frac{y_j}{p_j}; \quad \frac{\partial p_l}{\partial z_j} = \begin{cases} p_j(1-p_j) & l == j \\ -p_j p_l & l \neq j \end{cases}; \quad \frac{\partial \mathcal{L}}{\partial z_j} = p_j - y_j \quad (10)$$

$$\frac{\partial z_j}{\partial \kappa} = \mu_j^T \mathbf{x}; \quad \frac{\partial z_j}{\partial \mu_{jd}} = \kappa x_d; \quad \frac{\partial z_j}{\partial x_d} = \kappa \mu_{jd} \quad (11)$$

$$\frac{\partial x_d}{\partial f_d} = \begin{cases} \frac{\partial x_d}{\partial f_d} = \frac{\|\mathbf{f}\|^2 - f_d^2}{\|\mathbf{f}\|^3} = \frac{1-x_d^2}{\|\mathbf{f}\|} \\ \frac{\partial x_r}{\partial f_d} = \frac{-f_d x_r}{\|\mathbf{f}\|^3} = \frac{-x_d x_r}{\|\mathbf{f}\|} \end{cases} \quad (12)$$

$$\frac{\partial \mu_d}{\partial w_d} = \begin{cases} \frac{\partial \mu_d}{\partial w_d} = \frac{\|\mathbf{w}\|^2 - w_d^2}{\|\mathbf{w}\|^3} = \frac{1-\mu_d^2}{\|\mathbf{w}\|} \\ \frac{\partial \mu_r}{\partial w_d} = \frac{-w_d w_r}{\|\mathbf{w}\|^3} = \frac{-\mu_d \mu_r}{\|\mathbf{w}\|} \end{cases} \quad (13)$$

$$\frac{\partial \mathcal{L}}{\partial \kappa} = \sum_{j=1}^M (p_j - y_j) \mu_j^T \mathbf{x}; \quad \frac{\partial \mathcal{L}}{\partial \mu_{jd}} = (p_j - y_j) \kappa x_d \quad (14)$$

$$\frac{\partial \mathcal{L}}{\partial x_d} = \sum_{j=1}^M (p_j - y_j) \kappa \mu_{jd}; \quad \frac{\partial \mathcal{L}}{\partial f_d} = \frac{1}{\|\mathbf{f}\|} \left(\frac{\partial \mathcal{L}}{\partial x_d} - x_d \sum_r \frac{\partial \mathcal{L}}{\partial x_r} x_r \right) \quad (15)$$

3.3 Interpretation and discussion

The proposed SFR model represents each identity/class (*i.e.*, face) with the mean (μ) and concentration (κ) parameters of the vMF distribution, which (unlike weight and bias) express their direct *relationship* with the respective identity. μ provides an expected representation (*e.g.*, mean facial features) of the identity and κ (independently computed) indicates the variations within the samples from the identity. See Appendix A for the illustration from MNIST digits based experiment, where the generated images from μ_j effectively show the ability of vMF-FL to learn a representative images of the corresponding classes.

In terms of *discriminative feature learning* [39], [68], we can interpret the effectiveness of vMFML by analyzing the shape of the vMF distributions and vMFMMs in Figure 3 based on the κ value. While for high κ value the features are closely located around the mean direction μ , for low κ values they randomly spread around and can be located far from μ . We observe that κ also plays an important role to separate the vMFMM samples from different classes. A higher κ value will enforce the features to be more concentrated around μ to minimize intra-class variations (reduce angular distances of samples and mean) and maximize inter-class distances (see Figure 3 and 1(b)). Therefore, unlike [68] (jointly optimizes two losses: *softmax* and *center loss*), we⁴ can learn discriminative features by optimizing single loss function and save $M \times D$ parameters, where D is the features dimension.

The formulation of vMFML can naturally provide interpretation and its relationships with several concurrently proposed loss functions [38], [39], [48], [66]. In Eq. 9, by using a higher κ value for the true class compared to the rest, *i.e.*, $\kappa_{y_i} > \kappa_{j \neq y_i}$, vMFML can formulate the *large-margin softmax loss* [39] and *A(angular)-softmax loss* [38] under certain conditions⁵. The *L2-softmax loss* [48] is similar to vMFML

4. See Appendix B for further details.

5. See Appendix B for further details.

when weights $\|\mathbf{w}\| = 1$ and biases $b_{y_i} = 0$ are applied within its formulation. Indeed, this relationship is significant as it provides additional justification (why $\kappa\mu$ is important) for the vMFML proposal from the perspective of face image quality, see 4.3 for further details. The *reformulated softmax loss* proposed in NormFace [66] optimizes the cosine similarity and add a scaling parameter which is empirically found by analyzing the bound of softmax loss. Interestingly, this reformulation (Eq. 6 of [66]) is exactly similar to Eq. 9 and hence vMFML has an interesting interpretation from a different viewpoint. The above discussions clearly state that while vMFML is derived from a theoretical model, it can be explained based on the intuitions and empirical justifications presented in the concurrently proposed loss functions [38], [48], [66] for FR.

Normalization [12], [28], [51] becomes an increasingly popular technique to use within the CNN models. Our method (*s.t.* normalization in the final layer) takes the advantages of different normalization techniques due to its natural form of the features ($\|\mathbf{x}\| = 1$) and parameter ($\|\mu\| = 1$). Particularly, the term $\kappa\mu$ is equivalent to the reparameterization⁶ proposed by *weight normalization* [51] and $\mu^T \mathbf{x}$ is equivalent to the *cosine normalization* [12]. Both [51] and [12] provide their relationship with the *batch normalization* [28] under certain conditions, which can be equally applicable to our case.

3.4 Face Verification with the vMF-FL method

The proposed vMF-FL method learns discriminative facial feature representation from a set of 2D facial images. Therefore, we use it to extract facial features and verify pairs of face images [25], templates [30] and videos [69].

3.4.1 CNN model

In general, any CNN model can be used with the proposed vMF-FL method. In this work, we follow the recent trend [18], [24] and use a deeper CNN. To this aim, we choose the publicly available⁷ ResNet [24] based CNN model provided by the authors of [68]. It consists of 27 convolution (*Conv*), 4 pooling (*Pool*) and 1 fully connected (*FC*) layers. Figure 4 illustrates the CNN model, called *Res-27*. Each convolution uses a 3×3 kernel and is followed by a PReLU activation function. The CNN progresses from the lower to higher depth by decreasing the spatial resolution using a 2×2 *max Pool* layer while gradually increasing the number of feature maps from 32 to 512. The 512 dimensional output from the FC layer is then unit normalized which we consider as the desired directional features representation of the input 2D image. Finally, we use the proposed vMFML and optimize the CNN during training. Overall, the CNN comprises 36.1M parameters for feature representation and $(512 \times M) + 1$ parameters for the vMFML, where M is the total number of identities/classes in the training database. Note that, vMFML only requires one additional scalar parameter (κ) compared to the general softmax loss.

6. See Appendix B for further details.

7. Note that the CNN proposed in [68] is different than the publicly provided CNN by the same authors. Therefore, in order to avoid confusion, we do not cite our CNN model directly as [68].

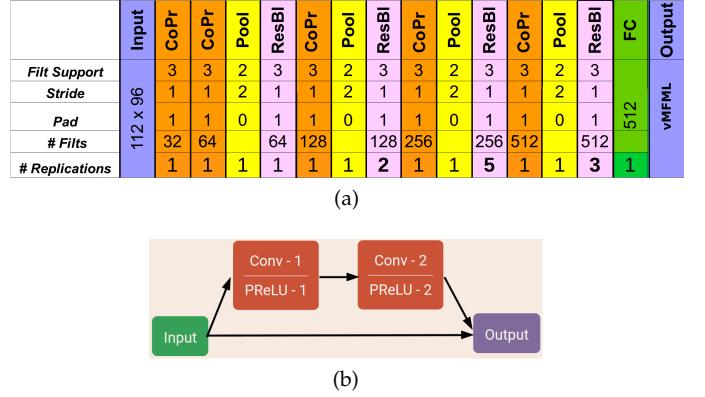


Fig. 4: (a) Illustration of the CNN model with vMFML. *CoPr* indicates *Convolution* followed by the *PReLU* activation. *ResBl* is a residual block [24] which computes $output = input + CoPr(CoPr(input))$. # *Replication* indicates how many times the same block is sequentially replicated. # *Filts* denotes the number of feature maps. (b) illustration of the residual block *ResBl*.

3.4.2 Face verification

Our face verification strategy follows the steps below:

- 1) *face image normalization*: we apply the following steps: (a) use the MTCNN [74] detector⁸ to detect the faces and landmarks; (b) normalize the face image by applying a 2D similarity transformation. The transformation parameters are computed from the location of the detected landmarks on the image and pre-set coordinates in a 112×96 image frame; (c) apply color to grayscale image conversion and (d) normalize the pixel intensity values within the range -1 to 1 . Note that, this normalization method is also applied during the training data preparation.
- 2) *features extraction*: use the CNN (trained with vMF-FL) to extract features from the original and horizontally flipped version and take the element-wise maximum value. For template [30] and video [69], obtain the features of an identity by taking element-wise average of the features from all of the images/frames.
- 3) *verification score computation*: compute the cosine similarity as the score and compare it to a threshold.

4 EXPERIMENTS, RESULTS AND DISCUSSION

We train the CNN model, use it to extract features and perform different scenarios for face verification, *i.e.*, single image-based [7], [25], multi-image or video-based [30], [69]. In order to verify the effectiveness of the proposed approach, we experiment on several datasets, namely LFW [25], IJB-A [30], YTF [69] and CACD [7], which impose various challenge to FR by collecting images from different sources and ensuring sufficient variations *w.r.t.* pose, illumination, occlusion, expression, resolution, age, geographic

8. In case of multiple faces, we take the face closer to the image center. If the landmarks detector fails we keep the face image by cropping it based on the detected bounding box.

regions, *etc.* Figure 6 provides examples of the pairs of images/videos/templates from different FR datasets.

LFW [25] dataset was particularly designed to study the problem of face verification with unconstrained facial images taken in everyday settings and collected from the web. It is considered as one of the most challenging and popular FR benchmark. YTF [69] is a dataset of unconstrained videos and designed to evaluate FR methods for matching faces in pairs of videos. Low image resolution, high illumination and pose variations on the YTF videos added significant challenges to the video based FR methods. IJB-A [30] is a recently proposed dataset, which raises the FR difficulty by collecting faces with high variations in pose, illumination, expression, resolution and occlusion. Unlike LFW, IJB-A includes faces which cannot be detected by standard face detectors and hence added significant amount of challenges. The CACD [7] dataset is specifically designed to study age invariant FR problem. It ensures large variations of the ages in order to add further challenge for the FR methods.

4.1 CNN Training

We collect the training images from the cleaned⁹ version of the MS-Celeb-1M [19] database, which consists of 4.61M images of 61.24K identities. In order to pre-process the training data we normalize (Sect. 3.4.2) the facial images of the dataset.

We train our CNN model using only the identity label of each image. We use 95% images (4.3M images) for training and 5% images (259K images) for monitoring and evaluating the loss and accuracy. We train our CNN using the *stochastic gradient descent* method and *momentum* set to 0.9. Moreover, we apply *L2 regularization* with the *weight decay* set to $5e^{-4}$. We begin the CNN training with a learning rate 0.1 for 2 epochs. Then we decrease it after each epoch by a factor 10. We stop the training after 5 epochs. We use 120 images in each mini-batch. During training, we apply data augmentation by horizontally flipping the images. Note that, during evaluation on a particular dataset, we do not apply any additional CNN training or fine-tuning or dimension reduction.

4.2 Results and Evaluation

First we evaluate the proposed *vMFML* by comparing it to the state-of-the-art loss functions. Next, we evaluate our *vMF-FL* based FR method on the most common and challenging FR benchmark datasets and compare it to the state-of-the-art FR methods.

4.2.1 Comparison of the Loss Functions

To gain insight on the effectiveness of the proposed loss function, we compare the proposed *vMFML* with some state-of-the-art loss functions using two different CNN architectures of different depth along with two training datasets of different size. Table 2 presents the results, where we use the LFW [25] face verification accuracy as a measure of the performance. We consider three commonly used loss

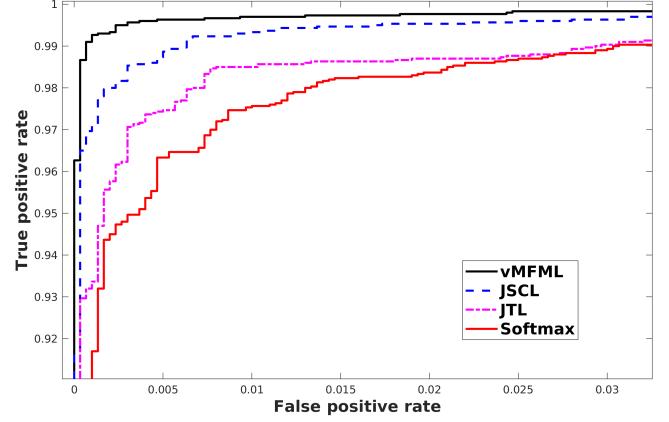


Fig. 5: Illustration of the ROC curves for different loss functions: *vMFML* Mixture Loss (*vMFML*), Softmax loss, joint *softmax+center* loss (*JSCL*) [68] and *softmax followed by triplet* loss (*STL*) [54]. Results obtained using the same CNN settings trained with the MS-Celeb-1M [19] dataset.

functions¹⁰ for FR, such as, Softmax loss, joint *softmax+center* loss (*JSCL*) [68] and *softmax followed by triplet* loss (*STL*) [54]. Note that, we use the same CNN model and best known training settings (learning rate, regularization, *etc.*) for individual loss function during the experiments.

First, we use the CASIA dataset [73] and train a shallower CNN proposed in [73], called *CasiaNet*. We observe that: *vMFML* (98%) > *STL* (97.93%) > *JSCL* (97.6%) > *softmax* (97.5%). Next, we train a deeper CNN, called *Res-27* (Section 3.4.1), with the same dataset and observe that: *vMFML* (99.18%) > *JSCL* (98.87%) > *STL* (98.13%) > *softmax* (97.4%). Therefore, *vMFML achieves better result (from 98% to 99.18%) with a deeper CNN*. Fig. 5 illustrates the corresponding receiver operating characteristic (ROC) curve for this comparison. Next, we train *Res-27* with the MS-Celeb-1M [19] and observe that: *vMFML* (99.63%) > *JSCL* (99.28%) > *STL* (98.83%) > *Softmax* (98.50%). That means, *vMFML improves its accuracy (from 99.18% to 99.63%) when trained with a larger dataset*. We also analyze the influence of a larger dataset by training *CasiaNet* with the MS-Celeb-1M [19] and observe that: *vMFML* (98.65%) > *STL* (98.41%) > *JSCL* (98.33%) > *Softmax* (98.21%). This observation further verifies the importance of a larger dataset to achieve better results with deep CNN, including the proposed *vMFML*.

The above analyses indicate that: 1) the *proposed vMFML* outperforms the state-of-the-art loss functions commonly used for FR; 2) *Res-27* is an appropriate CNN model and 3) the MS-Celeb-1M [19] is the best choice of training dataset to develop our FR method (see Sect. 3.4 and 4.1). In the next Section, we compare our FR method with the state-of-the-art for a variety of face verification tasks using different benchmarks. Besides, we conduct additional experiments and analysis to identify several common issues related to CNN training and discuss them in Sect. 4.3 .

4.2.2 Labeled Faces in the Wild (LFW)

The LFW dataset [25] is the *de facto benchmark* for evaluating unconstrained FR methods based on single image compar-

9. We take the list of 5.05M faces provided by [70] and keep non-overlapping (with test set) identities which have at least 30 images after successful landmarks detection.

10. Note that, we do not compare with the contrastive loss [11] as the joint *softmax+center* loss (*JSCL*) [68] has been shown to be more efficient than it.

TABLE 2: Comparison of different loss functions evaluated w.r.t. different CNN architectures and training datasets. The LFW [25] verification accuracy is used as a measure of the performance. Loss functions: vMF Mixture Loss (vMFML), Softmax loss, joint softmax+center loss (JSCL) [68] and softmax followed by triplet loss (STL) [54].

Database	CNN	Softmax	JSCL	STL	vMFML
Casia	CasiaNet	97.50	97.60	97.93	98.00
Casia	Res-27	97.40	98.87	98.13	99.18
MSCeleb-1M	Res-27	98.50	99.28	98.83	99.63
MSCeleb-1M	CasiaNet	98.21	98.33	98.42	98.65

ison. It consists of 13,233 images from 5,759 identities. Fig. 6(a) illustrates some face images from this dataset. The FR task requires verifying 6000 image pairs, which are equally divided into genuine and impostor pairs and comprises total 7.7K images from 4,281 identities. The LFW evaluation requires face verification in 10 pre-specified folds and report the average accuracy. It has different evaluation protocols and based on the recent trend we follow the *unrestricted-labeled-outside-data* protocol. Table 3 presents the results from our method in comparison with other state-of-the-art methods.

As can be seen in Table 3, our method achieves very competitive accuracy (99.63%) and ranks among the top performers, despite the fact that: (a) L2-Softmax [48] used¹¹ a 100 layers CNN model (3.7 times deeper/larger compared to us) to achieve 99.78%; (b) Baidu [36] combines 10 CNNs to obtain 99.77%, whereas we use a single CNN and (c) FaceNet [54] used 200M images of 8M identities to obtain 99.63%, whereas we train our CNN with only 4.51M images and 61.24K identities. It is interesting to note that, our LFW result (99.63%) is obtained by training CNN with the cleaned MSCeleb dataset that removes LFW overlapped identities. In removing the incorrectly labeled pairs within LFW (see Errata on the LFW webpage), our proposed vMF-FL further displays 99.68% accuracy. In light of the results from [36], [48], [55], [56], [57], [67], [78], the performance of the proposed method could be further improved by combining features from multiple CNN models or by using deeper (e.g., 100 layers) CNNs. However, Table 3 suggests saturation in the LFW results, as most of the methods already surpass human performance (97.53%) and they differ very few in terms of performance. Besides, it raises debate whether we should justify a method w.r.t. the real world FR scenario [35] based on the matching of only 6K pairs. To overcome this limitation, we follow the BLUFR LFW protocol [35], which exploits all LFW images, verifies 47M pairs per trial and provide the measure of true accept rate (TAR) at a low false accept rate (FAR). In Table 4, we provide our result for the verification rate (VR) at FAR=0.1% and compare it with the other methods which reported results in this protocol. As can be seen in Table 4, our method displays the best performance in comparison with the results of some state of the FR methods published so far.

Besides evaluating with different performance measures, we further challenge our proposed FR method on LFW from the age invariant perspective and benchmark it on

11. They achieved 99.60% when using the same CNN model that we used.

TABLE 3: Comparison of the state-of-the-art methods evaluated on the LFW benchmark [25].

FR method	# of CNNs	Dataset Info	Acc %
<i>vMF-FL (proposed)</i>	1	4.51M, 61.24K	99.63
Baidu [36]	10	1.2M, 1.8K	99.77
Baidu [36]	1	1.2M, 1.8K	99.13
L2-Softmax [48]	1	3.7, 58.2K	99.78
FaceNet [54]	1	200M, 8M	99.63
DeepVisage [20]	1	4.48M, 62K	99.62
RangeLoss [76]	1	1.5M, 100K	99.52
Sparse ConvNet [57]	25	0.29M, 12K	99.55
DeepID3 [55]	25	0.29M, 12K	99.53
Megvii [78]	4	5M, 0.2M	99.50
LF-CNNs [67]	25	0.7M, 17.2K	99.50
DeepID2+ [56]	25	0.29M, 12K	99.47
SphereFace [38]	1	0.49M, 10.57K	99.42
Center Loss [68]	1	0.7M, 17.2K	99.28
NormFace [66]	1	0.49M, 10.5K	99.19
MM-DFR [14]	8	0.49M, 10.57K	99.02
VGG Face [45]	1	2.6M, 2.6K	98.95
MFM-CNN [70]	1	5.1M, 79K	98.80
VIPLFaceNet [40]	1	0.49M, 10.57K	98.60
Webscale [59]	4	4.5M, 55K	98.37
AAL [72]	1	0.49M, 10.57K	98.30
FSS [65]	9	0.49M, 10.57K	98.20
Face-Aug-Pose-Syn [43]	1	2.4M, 10.57K	98.06
CASIA-Webface [73]	1	0.49M, 10.57K	97.73
Unconstrained FV [9]	1	0.49M, 10.5K	97.45
Deepface [58]	3	4.4M, 4K	97.35

TABLE 4: Comparison of the state-of-the-art methods based on the BLUFR LFW protocol [35].

Method	VR@FAR=0.1%
<i>vMF-FL(proposed)</i>	99.10
DeepVisage [20]	98.65
NormFace [66]	95.83
Center Loss [68]	92.97
FSS [65]	89.80
CASIA [73]	80.26

TABLE 5: Evaluation of the CALFW [77] dataset. Results from competitive methods are obtained from the paper [77].

	<i>vMF-FL (proposed)</i>	VGG-Face	Noisy Softmax
Acc%	94.20	86.50	82.52

the recently released variant of LFW, called Cross-Age LFW (CALFW) dataset [77]. CALFW is constructed by re-organizing (by crowdsourcing efforts) the LFW verification pairs with apparent age gaps (as large as possible) to form the positive pairs and then selecting negative pairs using individuals with the same race and gender (see Fig. 6(b) for some illustrations). Similar to LFW, CALFW evaluation consists of verifying 6000 pairs (equally divided into genuine and impostor pairs) of images in 10 pre-specified folds and report the average accuracy. Table 5 synthesizes the experimental result in comparison with the state of the art. As can be seen from Table 5, our method provides the best results compared to the other available results from different methods.

The results in Table 3, 4, 5 confirm the remarkable performance of *vMF-FL* on the LFW dataset. Next, we explore the FR task on videos and evaluate our method on the YouTube Faces [69] dataset.

TABLE 6: Comparison of the state-of-the-art methods evaluated on the Youtube Face [69]. *Ad.Tr.* denotes additional training is used.

FR method	Ad.Tr.	Accuracy (%)
<i>vMF-FL (proposed)</i>	N	96.46
DeepVisage [20]	N	96.34
L2-Softmax [48]	N	96.08
VGG Face [45]	N	91.60
Sparse ConvNet [57]	N	93.50
FaceNet [54]	N	95.18
SphereFace [38]	N	95.00
Center Loss [68]	N	94.90
NormFace [66]	Y	94.72
RangeLoss [76]	N	93.7
DeepID2+ [56]	N	93.20
MFM-CNN [70]	N	93.40
CASIA-Webface [73]	Y	92.24
Deepface [58]	Y	91.40
VGG Face [45]	Y	97.30
NAN [71]	Y	95.72

4.2.3 YouTube Faces (YTF)

The YTF database [69] (see Fig. 6(c) for some illustrations) is the *de facto benchmark* for evaluating the FR methods based on the unconstrained videos. It consists of 3,425 videos from 1,595 identities. Evaluation with YTF requires matching 5000 video pairs in 10 folds experiment and report the average accuracy. Each fold consists of 500 video pairs and ensures subject-mutually exclusive property. We follow the *restricted* protocol of YTF which restricts the access to only the similarity information. Table 6 provides the result from our method and compare it with the state-of-the-art methods.

Results in Table 6 show that our method provides the best accuracy (96.46%) in this dataset for the *restricted* protocol. In Table 6, we also report the results (separated with a horizontal line) from the *unrestricted* protocol, *i.e.*, access to similarity and identity ¹² information of the test data. Comparison indicates that our method is very competitive to the results from *unrestricted* protocol also. Moreover, by comparing the VGG Face [45] results with both protocols we observe that its accuracy increases significantly (from *restricted*-91.6% to *unrestricted*-97.3%) when the CNN is fine-tuned on the YTF dataset. Therefore, this comparison indicates that our result (96.46%) can be further enhanced if we train or fine-tune our CNN using the YTF data.

4.2.4 Cross-Age Celebrity Dataset (CACD)

The CACD dataset [7] aims to incorporate additional challenges to the FR task by explicitly focusing on the age invariant scenario (see Fig. 6(d) for some illustrations). Therefore, it ensures a large variations of the ages within the collected database of faces in the wild. CACD is constructed by collecting 163,446 images from 2000 identities, where the range of ages varies from 16 to 62. FR with the CACD dataset requires evaluating the similarity of 4000 pairs of images, which are equally splitted in ten folds experimental settings. The average from these ten folds accuracy is considered as

12. Access to identity information helps to create a large number of similar and dissimilar image pairs and hence can be further used to train or fine-tune CNN with an additional loss function, *e.g.*, the contrastive or the triplet loss [45].

TABLE 7: Comparison of the state-of-the-art methods evaluated on the CACD [7] dataset. VGG [45] result is obtained from [70].

FR method	Accuracy (%)
<i>vMF-FL (proposed)</i>	99.20
DeepVisage [20]	99.13
LF-CNNs [67]	98.50
MFM-CNN [70]	97.95
VGG Face [45]	96.00
CARC [7]	87.60
Human, Avg.	85.70
Human, Voting [7]	94.20

the measure of evaluation. In Table 7, we provide the results of *vMF-FL* and compare it with the state-of-the-art methods.

Results from Table 7 show that our method provides the best accuracy. Moreover, it is better than LF-CNN [67], which is a recent method specialized on age invariant face recognition.

4.2.5 IARPA Janus Benchmark A (IJB-A)

The recently proposed *IJB-A* database [30] is developed with the aim to augment more challenges to the FR task by collecting facial images with a wide variations in pose, illumination, expression, resolution and occlusion (see Fig. 6(e) for some illustrations). IJB-A is constructed by collecting 5,712 images and 2,085 videos from 500 identities, with an average of 11.4 images and 4.2 videos per identity. We follow the *compare* protocol of IJB-A because it measures the face verification accuracy. This protocol requires the comparison among pairs of templates, where each template consists of a set of images and video-frames. The evaluation protocol of IJB-A computes the true accept rate (TAR) at different fixed false accept rate (FAR), *e.g.*, 0.01 and 0.001. The evaluation requires computing the metrics in ten random split experimental settings. Note that we do not use the training data from each split and only evaluate on the test data using our once-trained FR method.

In Table 8, we present the results of *vMF-FL* and compare it with the state-of-the-art methods. We separate the results (with a horizontal line) to distinguish two categories of methods: (1) methods do not use IJB-A training data and evaluate test sets with a pre-trained CNN; our method belongs to this category and (2) methods use IJB-A training data and applies additional training, such as, CNN fine-tuning or metric learning.

From the results in Table 8, we observe that, while L2-softmax [48] provides the best result, our method provides competitive results among the others. We manually investigated our performance on IJB-A and observed that a large number of images were incorrectly pre-processed because of the failure of face and landmarks detection performance by the pre-processor. This is due to the fact that a large part of this dataset consists of very high pose and occluded faces, see Fig. 6(e) for few examples. Therefore, we conjecture that our performance could be much more improved if we apply a better pre-processor for this challenging database.

Note that, there are numerous methods, such as TA [13], NAN [71] and TPE [52], which use the CNN features and incorporates additional learning method to improve the results. Therefore, features from *vMF-FL* can be used with them [13], [52], [71] to further improve the results.

TABLE 8: Comparison of the state-of-the-art methods evaluated on the IJB-A benchmark [30]. ‘-’ indicates the information for the entry is unavailable. Methods which incorporates external training (ExTr) or CNN fine-tuning (FT) with IJB-A training data are separated with a horizontal line. VGG-Face result was provided by [52]. T@F denotes the *True Accept Rate at a fixed False Accept Rate (TAR@FAR)*.

FR method	ExTr	FT	T@F 0.01	T@F 0.001
<i>vMF-FL (proposed)</i>	N	N	0.897	0.850
L2-softmax [48]	N	N	0.968	0.938
DeepVisage [20]	N	N	0.887	0.824
VGG Face [45]	N	N	0.805	0.604
Face-Aug-Pose-Syn [43]	N	N	0.886	0.725
Deep Multipose [1]	N	N	0.787	-
Pose aware FR [42]	N	N	0.826	0.652
TPE [53]	N	N	0.871	0.766
All-In-One [49]	N	N	0.893	0.787
L2-softmax [48]	Y	N	0.970	0.943
All-In-One [49] + TPE	Y	N	0.922	0.823
Sparse ConvNet [57]	Y	N	0.726	0.460
FSS [65]	N	Y	0.729	0.510
TPE [53]	Y	N	0.900	0.813
Unconstrained FV [9]	Y	Y	0.838	-
TSE [52]	Y	Y	0.790	0.590
NAN [71]	Y	N	0.941	0.881
TA [13]	Y	N	0.939	0.836
End-To-End [10]	N	Y	0.787	-

4.2.6 Summary of the experimental results

Results of *vMF-FL* on different datasets prove that besides achieving significant results it generalizes very well and overcomes several difficulties which make unconstrained FR a challenging task.

4.3 Additional Analysis and Discussion

In this section, we perform in-depth analysis to further provide insight of the proposed method. We first study the sensitivity parameters of the proposed vMFML parameters (Sect. 4.3.1); Then, we conduct further experiments and analysis to discuss the influences of several CNN training related aspects, such as: (a) training datasets size (Sect. 4.3.2); (b) activation functions (Sect 4.3.3) and (c) normalization (Sect. 4.3.4). Finally, we also analyze the limitations of the proposed method (Sect. 4.3.5). In order to evaluate, we observe the accuracy and TAR@FAR=0.01 on the LFW benchmark.

4.3.1 Sensitivity analysis of the vMFML parameters

The proposed vMFML features two parameters, namely the concentration κ and the mean μ , whose sensitivity is analyzed here. In general, we can initialize κ with a small value (*e.g.*, 1) and learn it via backpropagation. However, we observe that the trained κ may provide sub-optimal results, especially when it is trained with large number of CNN parameters and updated with the same learning rate. To overcome this, we set κ to an approximated value as $\kappa = \sqrt{d/2}$ and set its learning rate by multiplying the CNN learning rate with a small value 0.001. An alternative choice is to set κ to a fixed value for the entire training period, where the value will be determined empirically. Our best results are achieved with $\kappa = 16$. On the other hand, the parameter μ does not exhibit any particular sensitivity and learned in a similar way to other CNN parameters.

TABLE 9: Analysis of the influences from training databases with different sizes and numbers of classes. T@F denotes the *True Accept Rate at a fixed False Accept Rate (TAR@FAR)*.

Min samp/id	Size, Class	Acc %	T@F 0.01
MSCeleb-1M [19]			
10	4.68M, 62.1K	99.57	0.9963
30	4.61M, 61.2K	99.63	0.9971
50	3.91M, 46.7K	99.55	0.9960
70	3.11M, 32.1K	99.53	0.9956
100	1.53M, 12.5K	99.35	0.9928
CASIA [73]	0.43M, 10.6K	99.18	0.9902
Pose-CASIA [43]	1.26M, 10.6K	99.21	0.9907
VGG Faces [45]	1.6M, 2.6K	98.45	0.9787

4.3.2 Impact of training dataset size

Training dataset size matters and plays important role to learn good feature representations with the deep CNN models [54], [78]. To investigate its effect on our FR method, first we train the *Res-27 CNN model* (Sect. 3.4.1) with vMFML (Sect. 3.2.2) using the well-known training datasets, such as the MSCeleb-1M [19], CASIA-Webface [73] and VGG Faces [45]. For the MSCeleb-1M [19] dataset, we created different subsets by selecting certain minimum number of images per identity. Experiments with these datasets and subsets help us to understand the learning capacity of the proposed FR method and identify the training dataset requirements to achieve better performance. Table 9 presents the results, which shows that the proposed method learns reasonably well facial representations from a wide range of different sized datasets. Additional observations are as follows:

- performance does not improve significantly with the increase of number of images and identities. We achieved only 0.28% improvement by enlarging the number of images 3 times and number of identities 5 times (compare the results between MSCeleb-100 *min-samp/id* with MSCeleb-10 *min-samp/id*). Perhaps, it is important to ensure larger number of samples per identity (*e.g.*, 100).
- synthesized images does not help significantly. By comparing CASIA [73] with pose-augmented-CASIA (≈ 3 times larger dataset) [43], we observe only 0.03% accuracy improvement. Perhaps the synthesized image quality affects the results.
- number of identities is equally important as the number of images. We observe that the VGG Face [45] dataset provides lower accuracy compared to the CASIA [73] dataset despite having ≈ 4 times more number of images (but ≈ 4 times less number of identities).

Therefore, the key finding from this experiment is that, while we can achieve reasonably good and comparable results with relatively smaller datasets (CASIA and MSCeleb-100 *min-samp/id*), we can achieve better results by training with larger dataset which provides sufficient number of images per identity.

4.3.3 Influence of activation function

In order to observe the influence of *activation functions*, we also used the ReLU activation instead of PReLU and observe that it decreases the accuracy by approximately 0.4%.

4.3.4 Feature normalization

Feature normalization [20], [28], [66] plays a significant role in the performance of deep CNN models. As discussed in Section 3.3, vMFML naturally integrates features normalization due to the SFR model (Sect. 3.1.1) and the vMF distribution (Sect. 3.2.1). The features of our FR method are unit-normalized vectors, *i.e.*, $\|\mathbf{x}\| = 1$. We observed that when learned with the proposed vMFML these normalized features provides significantly better results than the unnormalized features learned with the Softmax loss, see Table 2 for the performance comparison of *Softmax* vs *vMFML*.

The promising results achieved by the unit-normalized features (learned with vMFML) naturally raises the question - *can the unit-normalized features improves the accuracy with any loss function?* To answer this question, we train the CNN with the unit-normalized features $\|\mathbf{x}\| = 1$ and optimize the softmax loss under different settings in order to compare them w.r.t. vMFML. To explain the settings, we exploit the terms within the exponential of the nominators, *i.e.*, $\mathbf{w}^T \mathbf{x} + b_{y_i}$ (Softmax) with $\kappa \mu^T \mathbf{x}$ (vMFML). Our observations are:

- $\mathbf{w}^T \mathbf{x} + b_{y_i}$: provides very poor results, because the CNN training fails to converge as it get stuck at arbitrary local minimas even if with a range of different learning rates (from 0.1 to 0.0001) and with/without applying the *L2* regularization. Compare to the vMFML, the expectation from this setting is that the learned \mathbf{w} values are similar to the values observed from $\kappa \mu^T$, such that, $\|\mathbf{w}\| = \kappa$. That means, this setting verifies whether the learned weights can absorb κ within it. However, as the observed results suggest, it fails to do so.
- $\mathbf{w}^T (\mathbf{x} / \kappa) + b_{y_i}$: CNN successfully trains and provides results which is closer to the vMFML. For this modified Softmax κ acts as a scalar multiplier, whereas for vMFML κ signifies the shared concentration parameter value. Interestingly, this modified Softmax is the same as the *L2-Softmax* loss recently proposed by [48], which is motivated by the fact that the *L2*-norm of the features (learned with the Softmax loss) provides interesting information of the face image quality and attributes. Intuitively, the idea is to provide same attention to all face images regardless of their quality. Moreover, they interpret the multiplier (here κ) as a constraint on the features to lie on a hypersphere of a fixed radius. Note that, if bias is ignored then the difference among vMFML and *L2-Softmax* remains as \mathbf{w} vs μ . Besides, while the softmax loss imposes no constraints¹³ on the weights \mathbf{w} , vMFML applies natural constraint on μ , *i.e.*, $\|\mu\| = 1$.

Therefore, the key finding from the above observations is that the unit-normalized features are not sufficient alone and concentration parameter κ of vMFML plays significant role to efficiently learn the CNN models.

4.3.5 Limitations of the proposed method

Finally, we investigate the limitation of the proposed method on different datasets by observing the face image

¹³ Except L2 regularization which is an explicit settings for many ML problems

pairs for which the verification results are incorrect. Table 10 provides the information about the number and type of incorrect cases, which indicate a higher ratio of false rejection compared to false acceptance. Figure 6 provides illustrations of the top (selected based on the distance from threshold) examples of failure cases on different datasets. From an in-depth analysis of the erroneous results, our observations are as follows:

- On the LFW [25] failure cases, occlusion, variation of illumination and poor image resolution played important role. From the erroneous CALFW [77] pairs, we observe that poor image resolution appears as a common property. Besides, the false rejected pairs are suffering from high age difference.
- Failure cases on the YTF [69] dataset can be characterized by variations of illumination and poor image resolution. Besides, high pose variation plays an important role.
- Most of the CACD incorrect results occurred due to falsely rejecting the similar face image pairs, which indicate that our method encounters difficulties to recognize the same person when the age difference is large. Besides, we observe that the variations of illumination commonly appear in the incorrect results.
- The large number of errors on the IJB-A dataset can be characterized with high pose (mostly profile images, yaw angle more than 60 degrees) and poor image resolution. Unfortunately, both of these two reasons cause the failure of face and landmarks detection, which forced us to leave a large number of images without applying any pre-processing. Note that, our training dataset do not have any image for which the pre-processor failed to detect face and landmarks. Therefore, our method may perform poorly in case of the failure of the pre-processor.

From the above observations, we can particularly focus on several issues to handle in future, such as: (a) image resolution; (b) extreme pose variation (c) lighting normalization and (d) occlusions.

Poor image resolution is a common issue among the failure cases on different datasets. This observation is similar to the recent research by Best-Rowden and Jain [4], which proposed a method to automatically predict the face image quality. They found that the results in the IJB-A [30] dataset can be improved by removing low-quality faces before computing the similarity scores. Therefore, we can apply this approach for certain FR cases where multiple images are available for each identity. However, this approach will not work when only a single image is available per identity or when all available images have poor quality. In order to address this, we can incorporate face image super resolution based technique [26] as a part of our pre-processor. The idea is to apply super-resolution to those images which have very low image quality score computed by techniques such as [4].

Large facial pose causes degradation of FR performance [27], [43]. It has been addressed in recent researches [1], [42], [43], which proposed to overcome this problem by frontalization [58] and creating a larger training dataset with synthesized facial images of a range of facial poses [1], [43]. However, the performance with above approaches

TABLE 10: Analysis of the number of incorrect results made by the proposed method on different FR datasets.

Dataset	# of Errors	# of False Accept	# of False Reject
LFW [25]	22	7	15
CALFW [77]	368	111	257
YTF [69]	197	13	184
CACD [7]	29	3	26
IJB-A [30]	855	133	722

depends on the quality of frontalization or synthesis. Recently [27] proposed a method to generate photo-realistic facial images. We can adopt this proposal within our pre-processor to overcome the large pose related issues.

In order to deal with the *occlusion* related problems we can adopt the deep feature interpolation based approach [61] to recover missing face attributes. Besides, to deal with eye-glass related occlusions, we can enlarge training images per identity by synthesizing faces with eye-glass using a generative method, such as [31].

In order to deal with the *lighting* related issues, we can adopt recent techniques, such as [34] for specular reflection problem and [75] for shadow related problems. The *large age difference* in the same person image pair reduces the performance of our method. One possible way to overcome this problem is by augmenting (age based facial image synthesis with generative methods [2], [61]) more training images with the variation of ages. However, we must ensure that the synthesized images remains photo-realistic.

5 CONCLUSION

We proposed a novel *directional deep-features* learning method for FR by exploiting the concept of model-based clustering and combining it with the deep CNN models. First, we used the vMF mixture model as the theoretical basis to propose a statistical feature representation (SFR) model. Next, we developed an effective directional features learning method called vMF-FL, which formulated a novel loss function called vMFML. It has several interesting properties, such as: (a) learns discriminative features; (b) subsumes different loss functions and normalization techniques and (c) interprets relationships among parameters and object features. Extensive experiments on face verification confirms the efficiency and generalizability of vMF-FL as it achieved very competitive and state-of-the-art results on the benchmark FR datasets. We foresee several future perspectives: (a) use the learned model to synthesize identity preserving faces and enhance training dataset and (b) explore SFR model with the generative adversarial network; and (c) apply it for other vision tasks (*e.g.*, scene analysis), other domains (*e.g.*, NLP, speech analysis) and other tasks (*e.g.* clustering). Moreover, by ignoring the *equal privilege assumption* one can further analyze the variations within a class/cluster, which can be interesting for *unsupervised* problems.

APPENDIX A

EXPERIMENTAL JUSTIFICATION OF THE SFR MODEL

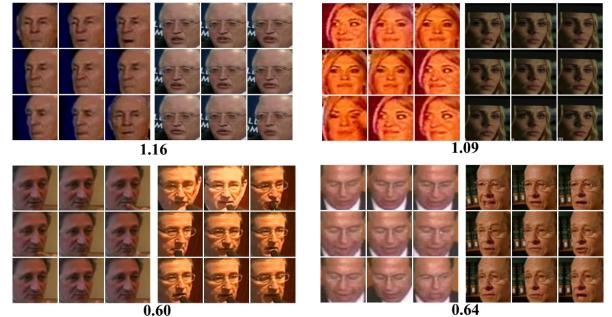
In this experiment, we experimentally justify the proposed SFR model (Sect. 3.1.1) and vMF-FL method (Sect. 3.1.2) by



(a) LFW [25], threshold distance 0.665



(b) CALFW [77], threshold distance 0.66



(c) YTF [69], threshold distance 0.72



(d) CACD [7], threshold distance 0.61



(e) IJB-A [30], threshold distance 0.49

Fig. 6: Illustration of the errors made by our method on different datasets: (a) LFW [25]; (b) CALFW [77]; (c) YTF [69]; (d) CACD [7] and (e) IJB-A [30]. The top row of each sub-figure provides examples of the false rejected pairs whereas the bottom row provides examples of the false accepted pairs.

exploiting the concept of deep convolutional autoencoders [29]. To conduct this experiment with the MNIST digits [33], we construct a deep autoencoder for this experiment, called vMF auto-encoder (vMF-AE). The *encoder/inverse-transformer* of vMF-FL consists of a 7 layers (6 convolution layers, 1 fully connected layer) deep architecture and the *decoder/transformer* of SFR model consists of a 4 layers (1 fully connected layer, 3 de-convolution layers) deep architecture. Fig. 7 (a) illustrates the vMF-AE method and Fig. 7 (b) shows the generated samples from the SFR-MNIST model (column 4-11).

vMF-AE combines the proposed SFR model with the vMF-FL method and learns in two steps. First it uses the vMF-FL method/block to simultaneously learn discriminative features (*vMF feature*) representation and the vMF Mixture Model (*vMFMM*) by optimizing the vMF mixture loss (Sect 3.2.2). Next, it learns the *decoder/transformer* of the SFR model to generate sample images from the features (learned by vMF-ML method) by optimizing pixel-wise binary-cross-entropy loss. When both learning tasks are done, we can generate 2D images of digits by sampling features from the learned vMFMM. While the column-3 of Fig. 7 (b) shows the images generated from different μ_j (*learned vMFMM parameters*), column 4-11 of Fig. 7 (b) show several 2D images generated from the sampled vMFMM features. We believe that, these illustrations demonstrate the originality of the proposed SFR model and hence provides additional justification to the proposed vMF-FL method.

APPENDIX B RELATIONSHIP WITH DIFFERENT LOSS FUNCTIONS AND NORMALIZATION METHOD

Center Loss [68] aims to enhance feature discrimination by minimizing the intra-class *mean-squared* distances. It has the following form [68]:

$$\mathcal{L}_{Center} = \frac{1}{2} \sum_{i=1}^N \|\mathbf{f}_i - \mathbf{c}_{y_i}\|^2 \quad (16)$$

where, \mathbf{f}_i and y_i are the features and ground-truth class labels of the i^{th} sample, \mathbf{c}_{y_i} is the center of class y_i . By comparing Eq. 16 and Eq. 9, we see that vMFML incorporates the *cosine/angular* similarity with the term $\mu_{y_i}^T \mathbf{x}_i$, where $\mu_{y_i} = \frac{\mathbf{c}_{y_i}}{\|\mathbf{c}_{y_i}\|}$ and $\mathbf{x} = \frac{\mathbf{f}}{\|\mathbf{f}\|}$, and use it to compute the posterior probability followed by computing the loss. Note that, the higher the cosine similarity, the higher the probability and hence the lower the loss. In a different way, we can say that the loss in Eq. 9 is minimized when the cosine similarity among the sample \mathbf{x}_i and the mean μ_{y_i} of its true class y_i is maximized. This indicates that, vMFML minimizes the intra-class distance by incorporating the distance computation task within its formulation.

The center loss is used as a supplementary loss to the softmax loss and the CNN learning task is achieved by joint (*softmax+center*) optimization. On the other hand, the above comparison among vMFML and both *softmax* loss and *center* loss reveals that vMFML can take the advantage of both with a single loss function and save $M \times D$ parameters, where D is the features dimension and M is the number of classes in the training dataset. These additional parameters

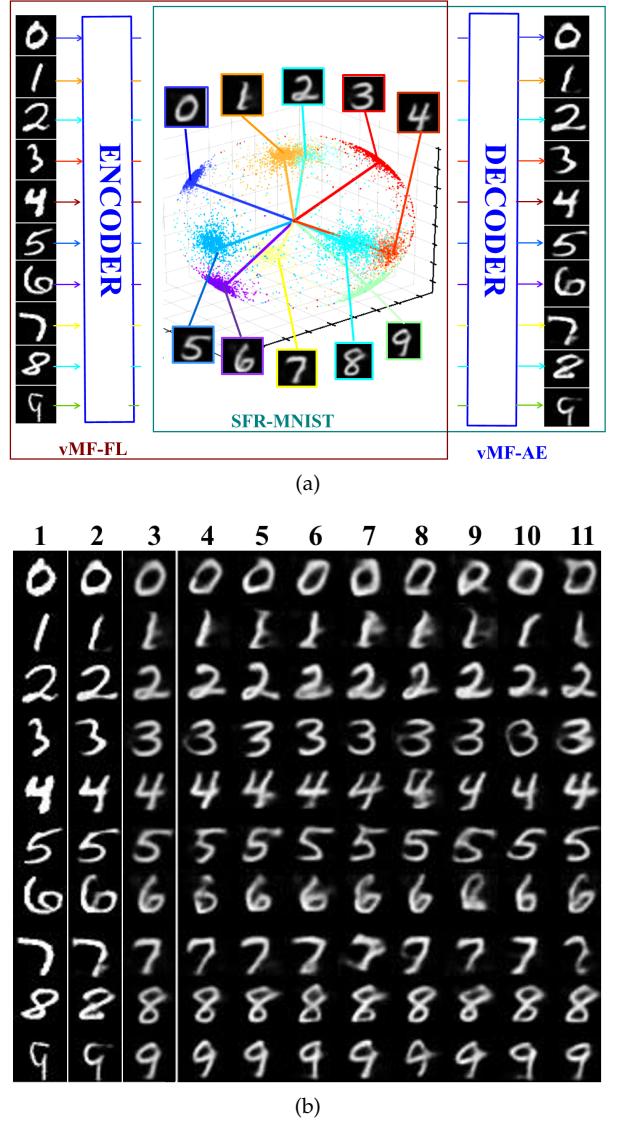


Fig. 7: (a) illustration of the vMF-AE method with MNIST digits, where the vMF-FL (*encoder/inverse-transformer*) learns discriminative 3D features and vMFMM for 10 digits-classes. In the 3D plot each dot represent features and each line represent the mean (μ_j) of respective class j . The images around the 3D sphere illustrates the generated images from respective (μ_j) and (b) illustration of the 2D images generated by the *transformer/decoder* of the SFR model, where col-1 represents the original image, col-2 shows the image generated from the encoded features of the original image, col-3 shows the image generated from the vMF mean (μ_j) of each class j , col-4-11 show 8 images from each digit-class j generated from the vMF samples (sampled using the mean (μ_j) and concentration (κ_j) of each class).

are used by the *center* loss to externally learn and save the centers.

Large-Margin Softmax Loss (LMSL) [39] is defined as:

$$\mathcal{L}_{LMSL} = - \sum_{i=1}^N \log \left(\frac{e^{\|\mathbf{w}_{y_i}\| \|\mathbf{f}_i\| \psi(\theta_{y_i})}}{e^{\|\mathbf{w}_{y_i}\| \|\mathbf{f}_i\| \psi(\theta_{y_i})} + \sum_{j \neq y_i} e^{\|\mathbf{w}_j\| \|\mathbf{f}_i\| \cos(\theta_j)}} \right) \quad (17)$$

where, $\psi(\theta_{y_i}) = (-1)^h \cos(m\theta) - 2h$ with integer $h \in$

$[0, m - 1]$ and $\theta \in \left[\frac{h\pi}{m}, \frac{(h+1)\pi}{m} \right]$. m denotes the margin, \mathbf{f}_i is the i^{th} image features, y_i is the true class label, \mathbf{w}_j is the weight corresponding to the j^{th} class and θ_j is the angle between \mathbf{w}_j and \mathbf{f}_i .

vMFML has similarity to LMSL when $\|\mathbf{w}_{y_i}\| = \|\mu\| = 1$, $\forall y_i$ and $\|\mathbf{f}_i\| = \|\mathbf{x}_i\| = 1$. In this condition, LMSL requires $m\theta_{y_i} < \theta_j$ ($j \neq y_i$), i.e., the angle between the sample and its true class is smaller than the rest of the classes subject to a margin multiplier m . With vMFML, this can be achieved by using a higher κ value (or multiply κ with a scalar multiplier m) for the true class compared to the rest, i.e., $\kappa_{y_i} > \kappa_{j \neq y_i}$. By considering the multiplier is equivalent to the margin m , we can rewrite Eq. 9 as:

$$\mathcal{L}_{vMFML} = - \sum_{i=1}^N \log \left(\frac{e^{m\kappa\mu_{y_i}^T \mathbf{x}_i}}{e^{m\kappa\mu_{y_i}^T \mathbf{x}_i} + \sum_{j \neq y_i} e^{\kappa\mu_j^T \mathbf{x}_i}} \right) \quad (18)$$

Note that, we do not use the Eq. 18 in this work, because it does not strictly follow the statistical features representation model proposed in this paper. The A (angular)-softmax loss [38] is a recent extension of the LMSL, which replaces the notion of *margin* with *angular margin*, considers $\|\mathbf{w}_{y_i}\| = 1$ and provides an equivalent loss formulation as Eq. 17. Therefore, Eq. 18 provides the relationship among vMFML and A-softmax [38] in a similar way as LMSL.

Weight Normalization [51] proposed a reparameterization of the standard weight vector \mathbf{w} as:

$$\mathbf{w} = g \frac{\mathbf{v}}{\|\mathbf{v}\|} \quad (19)$$

where, \mathbf{v} is a vector, g is a scalar and $\|\mathbf{v}\|$ is the norm of \mathbf{v} . This is related to the vMFML by considering the weight vector \mathbf{w} as:

$$\mathbf{w} = \kappa \frac{\mu}{\|\mu\|} = \kappa\mu; \quad [\|\mu\| = 1] \quad (20)$$

The comparison of Eq. 19 and Eq. 20 reveals that vMFML incorporates the properties of weight normalization subject to normalizing the activations of the CNN layer which are considered as the features.

REFERENCES

- [1] W. AbdAlmageed, Y. Wu, S. Rawls, S. Harel, T. Hassner, I. Masi, J. Choi, J. Lekust, J. Kim, and P. Natarajan. Face recognition using deep multi-pose representations. In *IEEE WACV*, pages 1–9, 2016.
- [2] G. Antipov, M. Baccouche, and J.-L. Dugelay. Face aging with conditional generative adversarial networks. *arXiv preprint arXiv:1702.01983*, 2017.
- [3] A. Banerjee, I. S. Dhillon, J. Ghosh, and S. Sra. Clustering on the unit hypersphere using von Mises-Fisher distributions. *Journal of Machine Learning Research*, 6(Sep):1345–1382, 2005.
- [4] L. Best-Rowden and A. K. Jaín. Automatic face image quality prediction. *arXiv preprint arXiv:1706.09887*, 2017.
- [5] A. Bhalerao and C.-F. Westin. Hyperspherical von mises-fisher mixture (HvMF) modelling of high angular resolution diffusion MRI. In *Proc. of MICCAI*, pages 236–243. Springer, 2007.
- [6] C. Biernacki, G. Celeux, and G. Govaert. Assessing a mixture model for clustering with the integrated completed likelihood. *IEEE TPAMI*, 22(7):719–725, 2000.
- [7] B.-C. Chen, C.-S. Chen, and W. H. Hsu. Face recognition and retrieval using cross-age reference coding with cross-age celebrity dataset. *IEEE Trans. on Multimedia*, 17(6):804–815, 2015.
- [8] D. Chen, X. Cao, L. Wang, F. Wen, and J. Sun. Bayesian face revisited: A joint formulation. In *ECCV*, pages 566–579. 2012.
- [9] J.-C. Chen, V. M. Patel, and R. Chellappa. Unconstrained face verification using deep CNN features. In *2016 IEEE WACV*, pages 1–9, 2016.
- [10] J.-C. Chen, R. Ranjan, A. Kumar, C.-H. Chen, V. Patel, and R. Chellappa. An end-to-end system for unconstrained face verification with deep convolutional neural networks. In *Proc. of IEEE ICCV-W*, pages 118–126, 2015.
- [11] S. Chopra, R. Hadsell, and Y. LeCun. Learning a similarity metric discriminatively, with application to face verification. In *Proc. of IEEE CVPR*, pages 539–546, 2005.
- [12] L. Chunjie, Y. Qiang, et al. Cosine Normalization: Using cosine similarity instead of dot product in neural networks. *arXiv preprint arXiv:1702.05870*, 2017.
- [13] N. Crosswhite, J. Byrne, O. M. Parkhi, C. Stauffer, Q. Cao, and A. Zisserman. Template adaptation for face verification and identification. *arXiv:1603.03958*, 2016.
- [14] C. Ding and D. Tao. Robust face recognition via multimodal deep face representation. *IEEE Trans. on Multimedia*, 17(11):2049–2058, 2015.
- [15] B. Fernando, E. Fromont, D. Muselet, and M. Sebban. Supervised learning of Gaussian mixture models for visual vocabulary generation. *Pattern Recognition*, 45(2):897–907, 2012.
- [16] J. Glover, G. Bradski, and R. B. Rusu. Monte Carlo pose estimation with quaternion kernels and the bingham distribution. *Robotics: Science and Systems VII*, page 97, 2012.
- [17] S. Gopal and Y. Yang. Von mises-fisher clustering models. In *Proc. of ICML*, pages 154–162, 2014.
- [18] J. Gu, Z. Wang, J. Kuen, L. Ma, A. Shahroudy, B. Shuai, T. Liu, X. Wang, and G. Wang. Recent advances in convolutional neural networks. *arXiv:1512.07108*, 2015.
- [19] Y. Guo, L. Zhang, Y. Hu, X. He, and J. Gao. Ms-Celeb-1M: A dataset and benchmark for large-scale face recognition. *CoRR*, abs/1607.08221, 2016.
- [20] A. Hasnat, J. Bohne, J. Milgram, S. Gentric, and L. Chen. Deepvisage: Making face recognition simple yet with powerful generalization skills. In *Proc. of IEEE ICCV-W AMFG*, Oct 2017.
- [21] M. A. Hasnat, O. Alata, and A. Trémeau. Joint Color-Spatial-Directional clustering and Region Merging (JCSD-RM) for unsupervised RGB-D image segmentation. *IEEE TPAMI*, 38(11):2255–2268, 2016.
- [22] M. A. Hasnat, O. Alata, and A. Trémeau. Model-based hierarchical clustering with Bregman divergences and Fishers mixture model: application to depth image analysis. *Statistics and Computing*, 26(4):861–880, 2016.
- [23] K. He, X. Zhang, S. Ren, and J. Sun. Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. In *Proc. of IEEE CVPR*, pages 1026–1034, 2015.
- [24] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *Proc. of IEEE CVPR*, 2016.
- [25] G. B. Huang, M. Ramesh, T. Berg, and E. Learned-Miller. Labeled faces in the wild: A database for studying face recognition in unconstrained environments. Technical report, Technical Report 07-49, University of Massachusetts, Amherst, 2007.
- [26] H. Huang, R. He, Z. Sun, and T. Tan. Wavelet-srnet: A wavelet-based cnn for multi-scale face super resolution. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1689–1697, 2017.
- [27] R. Huang, S. Zhang, T. Li, and R. He. Beyond face rotation: Global and local perception gan for photorealistic and identity preserving frontal view synthesis. *arXiv preprint arXiv:1704.04086*, 2017.
- [28] S. Ioffe and C. Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *Proc. of ICML*, pages 448–456, 2015.
- [29] D. P. Kingma and M. Welling. Auto-encoding variational bayes. In *Proceedings of International Conference on Learning Representations (ICLR)*, 2014.
- [30] B. F. Klare, B. Klein, E. Taborsky, A. Blanton, J. Cheney, K. Allen, P. Grother, A. Mah, M. Burge, and A. K. Jain. Pushing the frontiers of unconstrained face detection and recognition: IARPA Janus Benchmark A. In *Proc. of IEEE CVPR*, pages 1931–1939, 2015.
- [31] A. B. L. Larsen, S. K. Sønderby, H. Larochelle, and O. Winther. Autoencoding beyond pixels using a learned similarity metric. *arXiv preprint arXiv:1512.09300*, 2015.
- [32] E. Learned-Miller, G. B. Huang, A. RoyChowdhury, H. Li, and G. Hua. Labeled faces in the wild: A survey. In *Advances in face detection and facial image analysis*, pages 189–248. Springer, 2016.
- [33] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner. Gradient-based learning applied to document recognition. *Proc. of the IEEE*, 86(11):2278–2324, 1998.
- [34] C. Li, S. Lin, K. Zhou, and K. Ikeuchi. Specular highlight removal

- in facial images. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3107–3116, 2017.
- [35] S. Liao, Z. Lei, D. Yi, and S. Z. Li. A benchmark study of large-scale unconstrained face recognition. In *Proc. of IEEE IJCB*, pages 1–8, 2014.
- [36] J. Liu, Y. Deng, and C. Huang. Targeting ultimate accuracy: Face recognition via deep embedding. *arXiv:1506.07310*, 2015.
- [37] M. Liu, B. C. Vemuri, S.-I. Amari, and F. Nielsen. Shape retrieval using hierarchical total Bregman soft clustering. *IEEE TPAMI*, 34(12):2407–2419, 2012.
- [38] W. Liu, Y. Wen, Z. Yu, M. Li, B. Raj, and L. Song. Sphereface: Deep hypersphere embedding for face recognition. In *Proc. of IEEE CVPR*. IEEE, 2017.
- [39] W. Liu, Y. Wen, Z. Yu, and M. Yang. Large-Margin Softmax Loss for convolutional neural networks. In *Proc. of ICML*, pages 507–516, 2016.
- [40] X. Liu, M. Kan, W. Wu, S. Shan, and X. Chen. VIPLFaceNet: An open source deep face recognition SDK. *arXiv:1609.03892*, 2016.
- [41] K. V. Mardia and P. E. Jupp. *Directional statistics*, volume 494. Wiley. com, 2009.
- [42] I. Masi, S. Rawls, G. Medioni, and P. Natarajan. Pose-aware face recognition in the wild. In *Proc. of IEEE CVPR*, pages 4838–4846, 2016.
- [43] I. Masi, A. Tran, T. Hassner, J. T. Leksut, and G. Medioni. Do We Really Need to Collect Millions of Faces for Effective Face Recognition? In *ECCV*, 2016.
- [44] K. P. Murphy. *Machine learning: a probabilistic perspective*. The MIT Press, 2012.
- [45] O. M. Parkhi, A. Vedaldi, and A. Zisserman. Deep face recognition. *Proc. of BMVC*, 1(3):6, 2015.
- [46] A. B. Patel, M. T. Nguyen, and R. Baraniuk. A probabilistic framework for deep learning. In *Proc. of NIPS*, pages 2558–2566, 2016.
- [47] A. Prati, S. Calderara, and R. Cucchiara. Using circular statistics for trajectory shape analysis. In *Proc. of IEEE CVPR*, pages 1–8. IEEE, 2008.
- [48] R. Ranjan, C. D. Castillo, and R. Chellappa. L2-constrained softmax loss for discriminative face verification. *arXiv preprint arXiv:1703.09507*, 2017.
- [49] R. Ranjan, S. Sankaranarayanan, C. D. Castillo, and R. Chellappa. An all-in-one convolutional neural network for face analysis. *arXiv:1611.00851*, 2016.
- [50] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, A. C. Berg, and L. Fei-Fei. ImageNet Large Scale Visual Recognition Challenge. *IJCV*, 115(3):211–252, 2015.
- [51] T. Salimans and D. P. Kingma. Weight Normalization: A simple reparameterization to accelerate training of deep neural networks. In *Proc. of NIPS*, pages 901–901, 2016.
- [52] S. Sankaranarayanan, A. Alavi, C. Castillo, and R. Chellappa. Triplet probabilistic embedding for face verification and clustering. *arXiv:1604.05417*, 2016.
- [53] S. Sankaranarayanan, A. Alavi, and R. Chellappa. Triplet similarity embedding for face verification. *arXiv:1602.03418*, 2016.
- [54] F. Schroff, D. Kalenichenko, and J. Philbin. Facenet: A unified embedding for face recognition and clustering. In *Proc. of IEEE CVPR*, 2015.
- [55] Y. Sun, D. Liang, X. Wang, and X. Tang. DeepID3: Face recognition with very deep neural networks. *arXiv:1502.00873*, 2015.
- [56] Y. Sun, X. Wang, and X. Tang. Deeply learned face representations are sparse, selective, and robust. In *Proc. of IEEE CVPR*, pages 2892–2900, 2015.
- [57] Y. Sun, X. Wang, and X. Tang. Sparsifying neural network connections for face recognition. In *Proc. of IEEE CVPR*, 2016.
- [58] Y. Taigman, M. Yang, M. Ranzato, and L. Wolf. Deepface: Closing the gap to human-level performance in face verification. In *Proc. of IEEE CVPR*, pages 1701–1708, 2014.
- [59] Y. Taigman, M. Yang, M. Ranzato, and L. Wolf. Web-scale training for face identification. In *Proc. of IEEE CVPR*, pages 2746–2754, 2015.
- [60] Z. Tüske, M. A. Tahir, R. Schlüter, and H. Ney. Integrating Gaussian mixtures into deep neural networks: softmax layer with hidden variables. In *Proc. of ICASSP*, pages 4285–4289. IEEE, 2015.
- [61] P. Upchurch, J. Gardner, K. Bala, R. Pless, N. Snavely, and K. Weinberger. Deep feature interpolation for image content changes. *arXiv preprint arXiv:1611.05507*, 2016.
- [62] A. Van den Oord and B. Schrauwen. Factoring variations in natural images with deep Gaussian mixture models. In *Proc. of NIPS*, pages 3518–3526, 2014.
- [63] E. Variani, E. McDermott, and G. Heigold. A Gaussian mixture model layer jointly optimized with discriminative features within a deep neural network architecture. In *Proc. of ICASSP*, pages 4270–4274. IEEE, 2015.
- [64] D. H. T. Vu and R. Haeb-Umbach. Blind speech separation employing directional statistics in an expectation maximization framework. In *Proc. of ICASSP*. IEEE, 2010.
- [65] D. Wang, C. Otto, and A. K. Jain. Face search at scale. *IEEE TPAMI*, 2016.
- [66] F. Wang, X. Xiang, J. Cheng, and A. L. Yuille. Normface: L2 hypersphere embedding for face verification. In *Proceedings of the 25th ACM international conference on Multimedia*. ACM, 2017.
- [67] Y. Wen, Z. Li, and Y. Qiao. Latent factor guided convolutional neural networks for age-invariant face recognition. In *Proc. of IEEE CVPR*, pages 4893–4901, 2016.
- [68] Y. Wen, K. Zhang, Z. Li, and Y. Qiao. A discriminative feature learning approach for deep face recognition. In *Proc. of ECCV*, pages 499–515. Springer, 2016.
- [69] L. Wolf, T. Hassner, and I. Maoz. Face recognition in unconstrained videos with matched background similarity. In *Proc. of IEEE CVPR*, pages 529–534, 2011.
- [70] X. Wu, R. He, Z. Sun, and T. Tan. A light CNN for deep face representation with noisy labels. *arXiv:1511.02683*, 2015.
- [71] J. Yang, P. Ren, D. Chen, F. Wen, H. Li, and G. Hua. Neural aggregation network for video face recognition. *arXiv:1603.05474*, 2016.
- [72] H. Ye, W. Shao, H. Wang, J. Ma, L. Wang, Y. Zheng, and X. Xue. Face recognition via active annotation and learning. In *Proc. of ACM Multimedia*, pages 1058–1062. ACM, 2016.
- [73] D. Yi, Z. Lei, S. Liao, and S. Z. Li. Learning face representation from scratch. *arXiv:1411.7923*, 2014.
- [74] K. Zhang, Z. Zhang, Z. Li, and Y. Qiao. Joint face detection and alignment using multitask cascaded convolutional networks. *IEEE Signal Processing Letters*, 23(10):1499–1503, Oct 2016.
- [75] W. Zhang, X. Zhao, J.-M. Morvan, and L. Chen. Improving shadow suppression for illumination robust face recognition. *arXiv preprint arXiv:1710.05073*, 2017.
- [76] X. Zhang, Z. Fang, Y. Wen, Z. Li, and Y. Qiao. Range loss for deep face recognition with long-tail. *arXiv:1611.08976*, 2016.
- [77] T. Zheng, W. Deng, and J. Hu. Cross-age lfw: A database for studying cross-age face recognition in unconstrained environments. *arXiv preprint arXiv:1708.08197*, 2017.
- [78] E. Zhou, Z. Cao, and Q. Yin. Naive-deep face recognition: Touching the limit of LFW benchmark or not? *arXiv:1501.04690*, 2015.