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

Abul Hasnat[1], Julien Bohné[2], Jonathan Milgram[2], Stéphane Gentric[2], and Liming Chen[1]

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

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 features learning achieves a discriminative learning, *i.e.*, compacting the instances of the same class while increasing the distance of instances from different classes, and subsumes a number of loss functions or deep learning practice, *e.g.*, normalization. The experiments carried out on face verification using 4 different challenging face datasets, *i.e.*, LFW, IJB-A, YouTube faces and CACD, show the effectiveness of the proposed approach, which displays very competitive and state-of-the-art results.

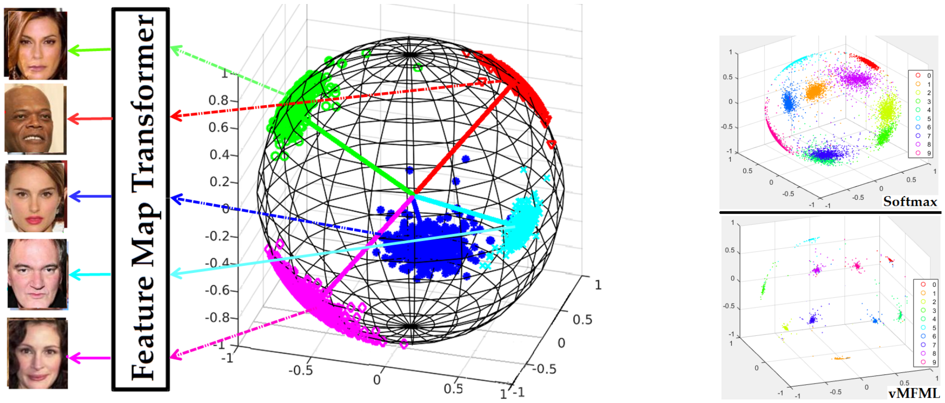
1 Introduction

A number of pattern recognition tasks, *e.g.*, face recognition, 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, many methods find useful to unit-normalize the final feature vectors, *e.g.*, [45, 10, 47, 51], and provide, beyond the simple softmax loss, additional or reinforced supervising signal, *e.g.*, center loss [47], large margin softmax loss [24], 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 in several computer vision tasks remains unclear from a theoretical viewpoint, which motivates us to *study the deep features 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 [28]. MM plays key role in model based clustering [4], which assumes a generative model, *i.e.*, each observation is a sample from a finite mixture of probability distributions. We adopt the *theoretical concept of MM to represent the 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 [12]. 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 unsupervised classification to analyze image [15] and text [2, 12]. In 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 with a 5 classes vMFMM. Figure 1(b) shows 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 [40, 36, 37, 30, 23] have surpassed human level performance. We used 4 different challenging face datasets, namely LFW [18] for face recognition in the wild, IJB-A [20] for face templates matching, YouTube Faces [48] (YTF) for video faces matching, and CACD [5] for cross age face matching. Using only one deep CNN-based neural network trained on the MS-Celeb dataset [14], the proposed method achieves 99.58% accuracy on LFW, 85% TAR@FAR=0.001 on IJB-A [20], 96.46% accuracy on YTF [48] and 99.2% accuracy on CACD [5]. These results indicate that our method achieves very competitive and state-of-the-art FR results, and generalizes very well across different datasets.



(a) (b)

Figure 1: (a) illustration of the proposed model, where *face* represent the *object* and (b) illustration of the 3D features learned from the MNIST digits [21], *top*: softmax loss, *bottom*: proposed (vMFML). Features from different classes are shown with different markers and colors in the sphere.

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

- we propose a *feature representation* model (Section 3.1.1) from a theoretical perspective. It is based on the statistical finite mixture model [28] with a directional distribution [25]. It provides a novel view for modeling the problem and better understanding the desired pattern recognition task. Therefore, it can help to develop efficient methods to achieve better results.

- we propose a *directional features representation learning* method, called vMF-FL (Section 3.1.2), which combines the theoretical model with the CNN model. vMF-FL provides a novel loss function, called vMFML (Section 3.2), whose formulation *w.r.t.* the backpropagation [21] method shows that it can be easily integrated with any CNN model. Moreover, vMFML is able to explain different loss functions [47, 24] and normalization methods [34, 8, 19]. vMFML not only interprets the relation among the parameters and features, but also improves CNN learning task *w.r.t.* efficiency (faster convergence) and performance (better accuracy). It can be used in a variety of classification task under the assumption of directional features.

- we demonstrate the proposed method for face verification and observe that it achieves state-of-the-art results and generalizes very well across different types of FR tasks.

In the remaining part, first 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

Mixture models (MM) [28] are relatively under-explored with the neural network (NN) based methods. [29] used the Gausian MM (GMM) to model deep NN as a mixture of transformers. [31] captured the variations of nuisance variables with the mixture of NNs. While both [28, 29] of them used MM within the NN, we consider NN externally and use as a single transformer. [43, 42] used GMM with NN and applied it for speech analysis. [43] learned discriminative features with their proposed GMM layer. [42] used the concept of log-linear model with GMM and NN. While our method is more similar to [43, 42], there are several differences: (a) we use directional (unit normalized) features; (b) we use the vMF [25] distribution which is more appropriate for directional features [16]; (c) our features representation model is based on a generative model-based [4] concept; and (d) we exploit the CNN [21] model and explore practical application of computer vision.

MM with directional distributions [25] have been used in a variety of domains to analyze images [16], speech [44], text [2, 12], gene expressions [2], shapes [32], pose [11], diffusion MRI [3], etc. However, they remain unexplored to learn discriminative features. In this paper, we aim to explore this by modeling the task with the vMF distribution [25, 2] and combining it with a CNN model. To our knowledge, *this is the first reported attempt to use a directional distribution with the CNN model*.

Loss functions are essential part of CNN training. Recent researches [47, 24] indicate that the widely used softmax loss cannot guarantee to provide discriminative features. [47] proposed center loss as a supplementary loss to minimize intra-class variations and hence to improve feature discrimination. We can achieve it with a single loss. [24] 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 (Section 3.3 provides more details).

Face recognition (FR) methods achieved remarkable results on the standard LFW [18] benchmark by using deep CNN models. FR methods commonly use the popular CNN models as their baseline model. For example, AlexNet is used by [33, 1, 26, 36] and VGGNet is used by [30, 9, 27, 1, 26, 10, 39]. *We use the ResNet [17] based deeper CNN model*. In general, FR methods use the softmax loss to train the CNN as an identity classifier. Recently the center loss [47] is proposed to enhance feature discrimination. Besides, several task specific loss functions, such as the contrastive loss [13, 7] is used by [38, 40, 39, 37, 50] and the triplet loss [36] is used by [36, 30, 9, 23]. The contrastive and triplet loss require careful preparation of the image pairs/triplets by maintaining certain constrains [36]. *vMF-FL simply learns features via identity classification and requires the class labels only*.

3 Methodology

In this section, first we present the *statistical features representation* model and then discuss the *features representation learning* method *w.r.t.* the model. Finally, we present the complete face recognition pipeline to demonstrate a practical application.

3.1 Model and Method

3.1.1 Statistical Features Representation (SFR) Model

We propose the SFR model based on the generative model-based approach [28], where the features are issued from a finite statistical mixture of probability distributions. Then, these features are transformed into the 2D image space using a transformer. Figure 1(a) (from right to left) provides an illustration, which considers a mixture of *von Mises-Fisher* (vMF) distributions [25] to model the features from different classes. The vMF distribution [25] 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} image features $\mathbf{x}_i$, let us call it ***vMF feature***, we define the SFR model with M classes as:

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

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 $V_d(.)$ 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 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 Features Learning (vMF-FL) Method

Figure 2 illustrates the workflow of the vMF-FL method, where the features are learned using an object identity classifier. The vMF-FL method consists of two sub-tasks: (1) mapping input 2D objects images to *vMF features* using the CNN model, which we use as the *transformer* and (2) classifying features to the respective classes based on the *discriminative* view of 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.

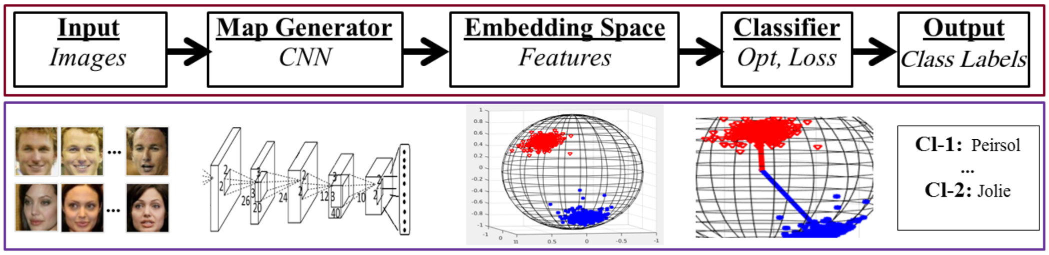


Figure 2: Workflow of the vMF-FL method. ***top:*** block diagram and ***bottom:*** view with an example.

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

Our proposed SFR model assumes that the 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 the formulation of vMFML.

vMF Mixture Model (vMFMM): For a d dimensional random unit vector $\mathbf{x} = [x_1, ..., x_d]^T \in S^{d-1} \subset \mathbb{R}^d$ (*i.e.*, $\|\mathbf{x}\|_2 = 1$), the density function of the vMF distribution is defined as [25]:

$$V_d(\mathbf{x}|\mu, \kappa) = C_d(\kappa)\ \exp(\kappa \mu^T \mathbf{x}) \qquad (2)$$

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, $I_\rho(.)$ is the modified Bessel function of the first kind. The top row of Figure 3 illustrates the vMF samples with different values of κ.

Let $\mathbf{X} = \{\mathbf{x}_i\}_{i=1,...,N}$ 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 defined as [2]: $g_v\left(\mathbf{x}_i|\Theta_M\right) = \sum_{j=1}^{M} \pi_j V_d\left(\mathbf{x}_i|\mu_j, \kappa_j\right)$, where $\Theta_M = \{(\pi_1, \mu_1, \kappa_1), ..., (\pi_M, \mu_M, \kappa_M)\}$ is the set of parameters, 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 Expectation Maximization (EM) method has been used to estimate the vMFMM parameters [2] by minimizing the negative log-likelihood value, *i.e.*, minimizing $-\,log(g_v\left(\mathbf{X}|\Theta_M\right))$. In the *E-step*, it estimates the posterior probability as: $p_{ij} = \frac{\pi_j\ V_d(\mathbf{x}_i|\mu_j,\kappa_j)}{\sum_{l=1}^{M} \pi_l\ V_d(\mathbf{x}_i|\mu_l,\kappa_l)}$, and in the *M-step* it updates parameters as: $\pi_j = \frac{1}{N}\sum_{i=1}^{N} p_{ij}$, $\mu_j = \frac{\sum_{i=1}^{N} p_{ij}\ \mathbf{x}_i}{\sum_{i=1}^{N} p_{ij}}$, $\bar{r} = \frac{\|\mu_j\|}{N\ \pi_j}$, $\mu_j = \frac{\mu_j}{\|\mu_j\|}$ and $\kappa_j = \frac{\bar{r}\ d - \bar{r}^3}{1-\bar{r}^2}$.

von Mises-Fisher Mixture Loss (vMFML) and optimization: Our vMF-FL method aims to learn discriminative features by minimizing the classification loss. Within this (*supervised classification*)

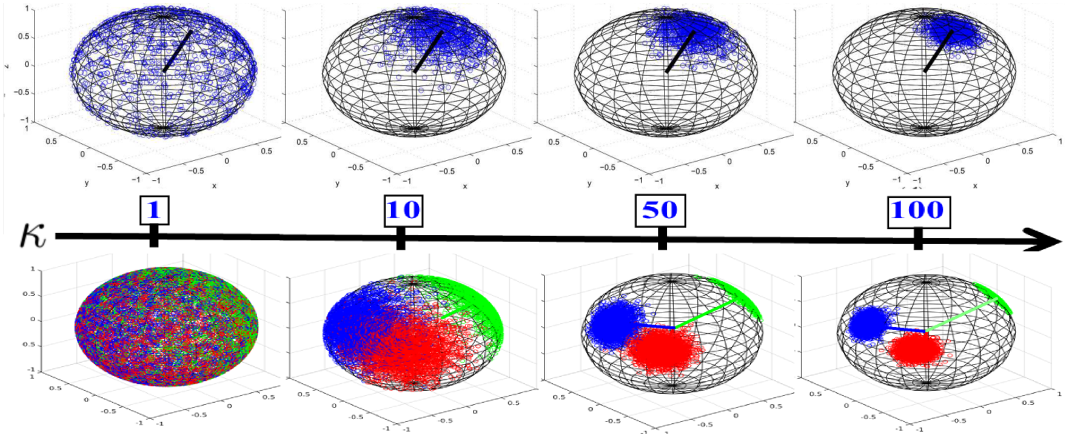


Figure 3: 3D directional samples from the vMF distribution (above arrow) and the vMFMM of 3 classes (below arrow). Samples are shown in the S^2 sphere for different values of κ.

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 SFR model as:

$$p_{ij} = \frac{\exp(\kappa \mu_j^T \mathbf{x}_i)}{\sum_{l=1}^{M} \exp(\kappa \mu_l^T \mathbf{x}_i)} \tag{3}$$

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

$$\mathcal{L}_{vMFML} = -\sum_{i=1}^{N}\sum_{j=1}^{M} y_{ij}\; log(p_{ij}) = -\sum_{i=1}^{N} log \frac{\exp(\kappa \mu_j^T \mathbf{x}_i)}{\sum_{l=1}^{M} \exp(\kappa \mu_l^T \mathbf{x}_i)} = -\sum_{i=1}^{N} log \frac{e^{z_{ij}}}{\sum_{l=1}^{M} e^{z_{il}}} \quad \left[z_j = \kappa \mu_j^T \mathbf{x}_i\right] \tag{4}$$

where, y_{ij} is the true class probability and we set $y_{ij} = 1$ as we only know the true class labels.

Now, we compare *vMFML* with *softmax loss*: $\mathcal{L}_{Softmax} = -\sum_{i=1}^{N} log \frac{exp(\mathbf{w}_j^T \mathbf{f}_i + b_j)}{\sum_{l=1}^{M} exp(\mathbf{w}_l^T \mathbf{f}_i + b_l)}$, where $\mathbf{f}_i$ is the i^{th} image features, $\mathbf{w}_j$ and b_j are the weights and bias of j^{th} class. 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. In a different way, we can say that *vMFML* provides an alternative form of the *softmax* loss by normalizing the weight and feature vectors and replacing the additive bias term with a multiplicative scalar term.

Now, we observe that the proposed vMF-FL method modifies the CNN training by replacing the softmax loss with 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 [21]. 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:

$$\frac{\partial z_j}{\partial \kappa} = \mu_j^T \mathbf{x}; \qquad \frac{\partial z_j}{\partial \mu_{jd}} = \kappa\, x_d; \qquad \frac{\partial z_j}{\partial x_d} = \kappa\, \mu_{jd} \tag{5}$$

$$\frac{\partial x_d}{\partial f_d} = \begin{cases} \frac{\partial x_d}{\partial f_d} = \frac{\|\mathbf{f}\|^2 - f_d^2}{\|\mathbf{f}\|^3} = \frac{1 - x_d^2}{\|\mathbf{f}\|} \\ \frac{\partial x_r}{\partial f_d} = \frac{-f_d f_r}{\|\mathbf{f}\|^3} = \frac{-x_d x_r}{\|\mathbf{f}\|} \end{cases}; \quad \frac{\partial \mu_d}{\partial w_d} = \begin{cases} \frac{\partial \mu_d}{\partial w_d} = \frac{\|\mathbf{w}\|^2 - w_d^2}{\|\mathbf{w}\|^3} = \frac{1 - \mu_d^2}{\|\mathbf{w}\|} \\ \frac{\partial \mu_r}{\partial w_d} = \frac{-w_d w_r}{\|\mathbf{w}\|^3} = \frac{-\mu_d \mu_r}{\|\mathbf{w}\|} \end{cases} \tag{6}$$

$$\frac{\partial \mathcal{L}}{\partial \kappa} = \sum_{j=1}^{M} (p_j - y_j)\, \mu_j^T\, \mathbf{x}\,; \qquad \frac{\partial \mathcal{L}}{\partial \mu_{jd}} = (p_j - y_j)\, \kappa\, x_d \tag{7}$$

$$\frac{\partial \mathcal{L}}{\partial x_d} = \sum_{j=1}^{M} (p_j - y_j)\, \kappa\, \mu_{jd}; \qquad \frac{\partial \mathcal{L}}{\partial f_d} = \frac{1}{\|\mathbf{f}\|} \left(\frac{\partial \mathcal{L}}{\partial x_d} - x_d \sum_{r} \frac{\partial \mathcal{L}}{\partial x_r} x_r \right) \tag{8}$$

3.3 Interpretation and discussion

The proposed model represents each class (*e.g.*, 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 class. μ provides an expected representation (*e.g.*, mean face image) of the class and κ (independently computed) indicates the variations within samples from the class.

In terms of *discriminative feature* learning [47, 24], 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. 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 samples appear to be uniformly distributed on the sphere. 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 [47] (jointly optimizes two losses), we can learn discriminative features by optimizing single loss function and save $M \times D$ parameters, where D is the features dimension. Moreover, in Eq. 4 by using a higher κ value for the true class compared to the rest, *i.e.*, $\kappa_{y_i} > \kappa_{j \neq y_i}$, our method can formulate the *large-margin softmax loss* [24] under certain condition.

Normalization [19, 34, 8] 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 reparameterization proposed by *weight normalization* [34] and $\mu^T \mathbf{x}$ is equivalent to the *cosine normalization* [8]. Both [34] and [8] provide their relationship with the *batch normalization* [19] 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 features from a set of 2D objects images. We use it to extract facial features and verify pairs of face images [18], templates [20] and videos [48].

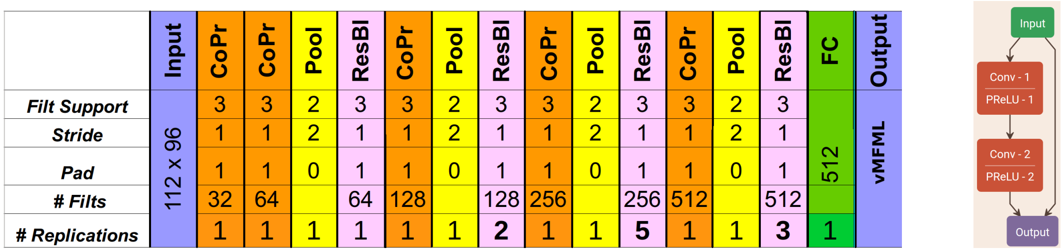
CNN model: In general, any CNN model can be used with the proposed vMF-FL method. In this work, we follow the recent trend [13, 17] to use a deeper CNN. To this aim, we choose the publicly available[1] CNN model provided by the authors of [47]. It consists of 27 convolution (*Conv*), 4 pooling (*Pool*) and 1 fully connected (*FC*) layers. Figure 4 illustrates the CNN model. 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 classes in the training database. Note that, vMFML only requires one additional scalar parameter (κ) compared to the general softmax loss.

Face verification: Our face verification strategy follows the steps below: ***1. pre-process***: normalize the face image by applying a 2D similarity transformation based on the detected facial landmarks (using MTCNN [52]) and pre-set coordinates in a 112×96 image frame; ***2. extract features***: 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 [20] and video [48], obtain the features of an identity by taking element-wise average of the features from all of the images/frames. and ***3. compute score***: 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 types (single-image [18, 5], multi-image or video [20, 48]) of face verification. In order to verify the effectiveness, we experiment on several datasets, namely LFW [18], IJB-A [20], YTF [48] and CACD [5], which impose various

[1] Note that the CNN proposed in [47] 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 [47].



	Input	CoPr	CoPr	Pool	ResBl	CoPr	Pool	ResBl	CoPr	Pool	ResBl	CoPr	Pool	ResBl	FC	Output
Filt Support	112 x 96	3	3	2	3	3	2	3	3	2	3	3	2	3	512	vMFML
Stride		1	1	2	1	1	2	1	1	2	1	1	2	1		
Pad		1	1	0	1	1	0	1	1	0	1	1	0	1		
# Filts		32	64		64	128		128	256		256	512		512		
# Replications		1	1	1	1	1	1	**2**	1	1	**5**	1	1	**3**	1	

(a)

(b)

Figure 4: (a) Illustration of the CNN model with vMFML. ***CoPr*** indicates *Convolution* followed by the *PReLU* activation. ***ResBl*** is a residual block [17] 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***.

challenge to FR by collecting images from different sources and ensuring sufficient variations *w.r.t.* pose, illumination, occlusion, expression, resolution, age, geographic regions, etc.

4.1 CNN Training

We collect the training images from the cleaned[2] version of the MS-Celeb-1M [14] database, which consists of 4.61M images of 61.24K identities. 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. 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 and dimension reduction.

4.2 Sensitivity Analysis

First, we analyze the sensitivity of the κ parameter. In general, we can initialize it with a small value (*e.g.*, 1) and learn it via backpropagation. However, we observe that the trained κ may provide sub-optimal results, specially 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.

Next, we observe the sensitivity *w.r.t.* CNN depth and training dataset size. First we use the CASIA dataset [51] and train a shallower CNN proposed in [51] and get 98% accuracy with vMFML, 97.6% with joint *softmax+center* loss (JSCL) and 97.5% with softmax loss. On the other hand, with the deeper CNN (Section 3.4) we achieved 99.1% with vMFML , 98.75% with JSCL and 97.4% with softmax loss. Therefore, vMFML achieves better result (from 98% to 99.1%) with a deeper CNN. Next, we train the deeper CNN with MS-Celeb-1M [14] and achieved 99.58% with vMFML , 99.28% with JSCL and 98.5% with softmax loss, which means vMFML improves result (from 99.1% to 99.58%) when trained with a larger dataset. We see that, JSCL shows similar sensitivity as vMFML.

4.3 Results and Evaluation

Now we evaluate the proposed *vMF-FL* method on the most common and challenging FR datasets.

[2] We take the list of 5.05M faces provided by [49] and keep non-overlapping (with test set) identities which has at least 30 images after successful landmarks detection.

Table 1: Comparison of the state-of-the-art methods evaluated on different datasets: LFW [18], YTF [48], CACD [5] and IJBA [20]. For a fair comparison, we consider the results from different methods which are obtained without any additional training (*e.g.*, metric learning, task specific training, etc.). If a method does not provide result without it, we indicate result with a '*' symbol. *T@F* means the TAR at a fixed FAR.

FR method	**# of CNNs**	**Tr. Data Info**	**LFW Acc %**	**YTF Acc %**	**CACD Acc%**	**IJB-A T@F 0.01**	**IJB-A T@F 0.001**
vMF-FL (proposed)	1	4.51M, 61.24K	99.58	***96.46***	***99.20***	***0.897***	***0.850***
Baidu [23]	10	1.2M, 1.8K	***99.77***	-	-	-	-
Baidu [23]	1	1.2M, 1.8K	99.13	-	-	-	-
FaceNet [36]	1	200M, 8M	99.63	95.18	-	-	-
Sparse CNet [39]	25	0.29M, 12K	99.55	93.5*	-	0.726*	0.460*
DeepID3 [37]	25	0.29M, 12K	99.53	-	-	-	-
Megvii [53]	4	5M, 0.2M	99.50	-	-	-	-
LF-CNNs [46]	25	0.7M, 17.2K	99.50	-	98.50	-	-
DeepID2+ [38]	25	0.29M, 12K	99.47	93.20	-	-	-
Center Loss [47]	1	0.7M, 17.2K	99.28	94.90	-	-	-
MM-DFR [10]	8	0.49M, 10.57K	99.02	-	-	-	-
VGG Face [30]	1	2.6M, 2.6K	98.95	91.60	96.00	0.805	0.604
MFM-CNN [49]	1	5.1M, 79K	98.80	93.40	97.95	-	-
L. M. S. Loss [24]	1	0.49M, 10.57K	98.71	-	-	-	-
FSS [45]	9	0.49M, 10.57K	98.2	-	-	0.729*	0.510*
Aug-Pose-Syn [27]	1	2.4M, 10.57K	98.06	-	-	0.886	0.725
Deepface [40]	3	4.4M, 4K	97.35	91.4	-	-	-
Unconst. FV [6]	1	0.49M, 10.5K	97.15	-	-	0.838*	-
CASIA [51]	1	0.49M, 10.57K	96.13	88.0	-	-	-
Deep Multipose [1]	6	2.4M, 10.5K	-	-	-	0.787	-
Pose aware FR [26]	5	2.4M, 10.5K	-	-	-	0.826	0.652
TPE [35]	1	0.49M, 10.57K	-	-	-	0.871	0.766
All-In-One [33]	1	0.49M, 10.57K	-	-	-	0.893	0.787

Labeled Faces in the Wild (LFW) [18] is one of the most popular and challenging dataset for evaluating unconstrained FR methods. The FR task requires verifying 6000 image pairs in 10 folds and report average accuracy. These pairs comprises 7.7K images of 4,281 identities. Based on the recent trend, we follow the *unrestricted-labeled-outside-data* protocol for evaluation. Results in Table 1 show that, our method achieves very competitive accuracy (99.58%) and among the top performers, despite the fact that: (a) we use single CNN, whereas Baidu [23] used 10 CNNs to obtain 99.77% and (b) we train CNN with comparatively much less amount of data and identities, whereas FaceNet [36] used 200M images of 8M identities to obtain 99.63%. Besides, results from [23, 39, 37, 53, 46, 38] indicate that we may further improve our result by combining features from multiple CNN models.

The results in the Table 1 indicates saturation, because all of the methods achieve close to or more than human performance (97.53%). Besides, it is argued that matching only 6K pairs is insufficient to justify a method *w.r.t.* the real world FR scenario [22]. Therefore, we follow the BLUFR LFW protocol [22] and measure the true accept rate (TAR) at a low false accept rate (FAR). BLUFR [22] protocol exploits all LFW images and verifies 47M pairs per trial. We compute the verification rate (VR) at FAR=0.1% and compare with the methods which reported results in this protocol. We observe that: ***vMF-FL*** (99.1) $>$ Center Loss[3] [47] (92.97%) $>$ FSS [45] (89.8%) $>$ CASIA [51] (80.26%), *i.e.*, our method is the best among the results published so far. Therefore, this result together with Table 1 confirm the remarkable performance of *vMF-FL* on the LFW.

YouTube Faces [48] (YTF) is a widely used FR dataset of unconstrained videos. It consists of 3,425 videos of 1,595 identities. YTF evaluation requires matching 5000 video pairs in 10 folds and report average accuracy. Each fold consists of 500 video pairs and ensures subject-mutually exclusive property. We follow the *restricted* protocol of YTF, *i.e.*, access to only the similarity information. Results in Table 1 show that our method provides the best accuracy (96.46%) in this dataset.

[3] Results computed from the features publicly provided by the authors.

Cross-Age Celebrity Dataset (CACD) [5] dataset aims to ensure large variations of the ages in the wild. It consists of 163,446 images of 2000 identities with the age ranging from 16 to 62. CACD evaluation requires verifying 4000 image pairs in ten folds and report average accuracy. Results from Table 1 show that our method provides the best accuracy. Moreover, it is better than LF-CNN [46], which is a recent method specialized on age invariant face recognition.

IARPA Janus Benchmark A (IJB-A) [20] database aims at raising the difficulty of FR by incorporating more variations in pose, illumination, expression, resolution and occlusion. It consists of 5,712 images and 2,085 videos of 500 identities. The FR task compares templates, which is a set of images and video-frames. The evaluation protocol requires computing TAR at different fixed FAR, *e.g.*, 0.01 and 0.001. From the results in Table 1, we observe that, our method provides the best results among the others. Note that, there are numerous methods, such as TA [9], NAN [50] and TPE [35], which use the CNN features and incorporates additional learning method to improve the results. Therefore, features from vMF-FL can be used with them [9, 50, 35] to further improve the 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.

5 Conclusion

We proposed a novel *directional deep-features* learning method by exploiting the concept of model-based clustering. 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 . 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 withing a class/cluster, which can be interesting for *unsupervised* problems.

References

[1] AbdAlmageed W, Wu Y, Rawls S, Harel S, Hassner T, Masi I, Choi J, Lekust J, Kim J, Natarajan P (2016) Face recognition using deep multi-pose representations. In: IEEE WACV, pp 1–9

[2] Banerjee A, Dhillon IS, Ghosh J, Sra S (2005) Clustering on the unit hypersphere using von Mises-Fisher distributions. Journal of Machine Learning Research 6(Sep):1345–1382

[3] Bhalerao A, Westin CF (2007) Hyperspherical von mises-fisher mixture (HvMF) modelling of high angular resolution diffusion MRI. In: Proc. of MICCAI, Springer, pp 236–243

[4] Biernacki C, Celeux G, Govaert G (2000) Assessing a mixture model for clustering with the integrated completed likelihood. IEEE TPAMI 22(7):719–725

[5] Chen BC, Chen CS, Hsu WH (2015) Face recognition and retrieval using cross-age reference coding with cross-age celebrity dataset. IEEE Trans on Multimedia 17(6):804–815

[6] Chen JC, Patel VM, Chellappa R (2016) Unconstrained face verification using deep cnn features. In: 2016 IEEE WACV, pp 1–9

[7] Chopra S, Hadsell R, LeCun Y (2005) Learning a similarity metric discriminatively, with application to face verification. In: Proc. of IEEE CVPR, pp 539–546

[8] Chunjie L, Qiang Y, et al (2017) Cosine Normalization: Using cosine similarity instead of dot product in neural networks. arXiv preprint arXiv:170205870

[9] Crosswhite N, Byrne J, Parkhi OM, Stauffer C, Cao Q, Zisserman A (2016) Template adaptation for face verification and identification. arXiv:160303958

[10] Ding C, Tao D (2015) Robust face recognition via multimodal deep face representation. IEEE Trans on Multimedia 17(11):2049–2058

[11] Glover J, Bradski G, Rusu RB (2012) Monte Carlo pose estimation with quaternion kernels and the bingham distribution. Robotics: Science and Systems VII p 97

[12] Gopal S, Yang Y (2014) Von mises-fisher clustering models. In: Proc. of ICML, pp 154–162

[13] Gu J, Wang Z, Kuen J, Ma L, Shahroudy A, Shuai B, Liu T, Wang X, Wang G (2015) Recent advances in convolutional neural networks. arXiv:151207108

[14] Guo Y, Zhang L, Hu Y, He X, Gao J (2016) Ms-Celeb-1M: A dataset and benchmark for large-scale face recognition. CoRR abs/1607.08221

[15] Hasnat MA, Alata O, Trémeau A (2016) Joint Color-Spatial-Directional clustering and Region Merging (JCSD-RM) for unsupervised RGB-D image segmentation. IEEE TPAMI 38(11):2255–2268

[16] Hasnat MA, Alata O, Trémeau A (2016) Model-based hierarchical clustering with Bregman divergences and Fishers mixture model: application to depth image analysis. Statistics and Computing 26(4):861–880

[17] He K, Zhang X, Ren S, Sun J (2016) Deep residual learning for image recognition. In: Proc. of IEEE CVPR

[18] Huang GB, Ramesh M, Berg T, Learned-Miller E (2007) Labeled faces in the wild: A database for studying face recognition in unconstrained environments. Tech. rep., Technical Report 07-49, University of Massachusetts, Amherst

[19] Ioffe S, Szegedy C (2015) Batch normalization: Accelerating deep network training by reducing internal covariate shift. In: Proc. of ICML, pp 448–456

[20] Klare BF, Klein B, Taborsky E, Blanton A, Cheney J, Allen K, Grother P, Mah A, Burge M, Jain AK (2015) Pushing the frontiers of unconstrained face detection and recognition: IARPA Janus Benchmark A. In: Proc. of IEEE CVPR, pp 1931–1939

[21] LeCun Y, Bottou L, Bengio Y, Haffner P (1998) Gradient-based learning applied to document recognition. Proc of the IEEE 86(11):2278–2324

[22] Liao S, Lei Z, Yi D, Li SZ (2014) A benchmark study of large-scale unconstrained face recognition. In: Proc. of IEEE IJCB, pp 1–8

[23] Liu J, Deng Y, Huang C (2015) Targeting ultimate accuracy: Face recognition via deep embedding. arXiv:150607310

[24] Liu W, Wen Y, Yu Z, Yang M (2016) Large-Margin Softmax Loss for convolutional neural networks. In: Proc. of ICML, pp 507–516

[25] Mardia KV, Jupp PE (2009) Directional statistics, vol 494. Wiley. com

[26] Masi I, Rawls S, Medioni G, Natarajan P (2016) Pose-aware face recognition in the wild. In: Proc. of IEEE CVPR, pp 4838–4846

[27] Masi I, Tran A, Hassner T, Leksut JT, Medioni G (2016) Do We Really Need to Collect Millions of Faces for Effective Face Recognition? In: ECCV

[28] Murphy KP (2012) Machine learning: a probabilistic perspective. The MIT Press

[29] Van den Oord A, Schrauwen B (2014) Factoring variations in natural images with deep Gaussian mixture models. In: Proc. of NIPS, pp 3518–3526

[30] Parkhi OM, Vedaldi A, Zisserman A (2015) Deep face recognition. Proc of BMVC 1(3):6

[31] Patel AB, Nguyen MT, Baraniuk R (2016) A probabilistic framework for deep learning. In: Proc. of NIPS, pp 2558–2566

[32] Prati A, Calderara S, Cucchiara R (2008) Using circular statistics for trajectory shape analysis. In: Proc. of IEEE CVPR, IEEE, pp 1–8

[33] Ranjan R, Sankaranarayanan S, Castillo CD, Chellappa R (2016) An all-in-one convolutional neural network for face analysis. arXiv:161100851

[34] Salimans T, Kingma DP (2016) Weight Normalization: A simple reparameterization to accelerate training of deep neural networks. In: Proc. of NIPS, pp 901–901

[35] Sankaranarayanan S, Alavi A, Castillo C, Chellappa R (2016) Triplet probabilistic embedding for face verification and clustering. arXiv:160405417

[36] Schroff F, Kalenichenko D, Philbin J (2015) Facenet: A unified embedding for face recognition and clustering. In: Proc. of IEEE CVPR

[37] Sun Y, Liang D, Wang X, Tang X (2015) DeepID3: Face recognition with very deep neural networks. arXiv:150200873

[38] Sun Y, Wang X, Tang X (2015) Deeply learned face representations are sparse, selective, and robust. In: Proc. of IEEE CVPR, pp 2892–2900

[39] Sun Y, Wang X, Tang X (2016) Sparsifying neural network connections for face recognition. In: Proc. of IEEE CVPR

[40] Taigman Y, Yang M, Ranzato M, Wolf L (2014) Deepface: Closing the gap to human-level performance in face verification. In: Proc. of IEEE CVPR, pp 1701–1708

[41] Taigman Y, Yang M, Ranzato M, Wolf L (2015) Web-scale training for face identification. In: Proc. of IEEE CVPR, pp 2746–2754

[42] Tüske Z, Tahir MA, Schlüter R, Ney H (2015) Integrating Gaussian mixtures into deep neural networks: softmax layer with hidden variables. In: Proc. of ICASSP, IEEE, pp 4285–4289

[43] Variani E, McDermott E, Heigold G (2015) A Gaussian mixture model layer jointly optimized with discriminative features within a deep neural network architecture. In: Proc. of ICASSP, IEEE, pp 4270–4274

[44] Vu DHT, Haeb-Umbach R (2010) Blind speech separation employing directional statistics in an expectation maximization framework. In: Proc. of ICASSP, IEEE

[45] Wang D, Otto C, Jain AK (2016) Face search at scale. IEEE TPAMI

[46] Wen Y, Li Z, Qiao Y (2016) Latent factor guided convolutional neural networks for age-invariant face recognition. In: Proc. of IEEE CVPR, pp 4893–4901

[47] Wen Y, Zhang K, Li Z, Qiao Y (2016) A discriminative feature learning approach for deep face recognition. In: Proc. of ECCV, Springer, pp 499–515

[48] Wolf L, Hassner T, Maoz I (2011) Face recognition in unconstrained videos with matched background similarity. In: Proc. of IEEE CVPR, pp 529–534

[49] Wu X, He R, Sun Z, Tan T (2015) A light CNN for deep face representation with noisy labels. arXiv:151102683

[50] Yang J, Ren P, Chen D, Wen F, Li H, Hua G (2016) Neural aggregation network for video face recognition. arXiv:160305474

[51] Yi D, Lei Z, Liao S, Li SZ (2014) Learning face representation from scratch. arXiv:14117923

[52] Zhang K, Zhang Z, Li Z, Qiao Y (2016) Joint face detection and alignment using multitask cascaded convolutional networks. IEEE Signal Processing Letters 23(10):1499–1503, DOI 10.1109/LSP.2016.2603342

[53] Zhou E, Cao Z, Yin Q (2015) Naive-deep face recognition: Touching the limit of LFW benchmark or not? arXiv:150104690