# COSONet: Compact Second-Order Network for Video Face Recognition

Yirong Mao[1,2][0000−0002−7603−4341], Ruiping Wang[1,2][0000−0003−1830−2595], Shiguang Shan[1,2][0000−0002−8348−392X], and Xilin Chen[1,2][0000−0003−3024−4404]

[1] Key Laboratory of Intelligent Information Processing of Chinese Academy of Sciences (CAS), Institute of Computing Technology, CAS, Beijing, 100190, China
[2] University of Chinese Academy of Sciences, Beijing, 100049, China
{yirong.mao}@vipl.ict.ac.cn, {wangruiping, sgshan, xlchen}@ict.ac.cn

**Abstract.** In this paper, we study the task of video face recognition. The face images in the video typically cover large variations in expression, lighting, or pose, and also suffer from video-type noises such as motion blur, out-of-focus blur and low resolution. To tackle these two types of challenges, we propose an extensive framework which contains three aspects: neural network design, training data augmentation, and loss function. First, we devise an expressive COmpact Second-Order network (COSONet) to extract features from faces with large variations. The network manages to encode the correlation (*e.g.* sample covariance matrix) of local features in a spatial invariant way, which is useful to model the global texture and appearance of face images. To further handle the curse of high-dimensional problem in the sample covariance matrix, we apply a layer named 2D fully connected (2D-FC) layer with few parameters to reduce the dimension. Second, due to no video-type noises in still face datasets and small inter-frame variation in video face datasets, we augment a large dataset with both large face variations and video-type noises from existing still face dataset. Finally, to get a discriminative face descriptor while balancing the effect of images with various quality, a mixture loss function which encourages the discriminability and simultaneously regularizes the feature is elaborately designed. Detailed experiments show that the proposed framework can achieve very competitive accuracy over state-of-the-art approaches on IJB-A and PaSC datasets.

**Keywords:** Video Face Recognition, Second-Order Network, Data Augmentation

## 1 Introduction

As tremendous video data are being created from real-word application scenarios such as video surveillance, movies, or daily photo albums, video face recognition (VFR) has caught much more attention nowadays.

To solve this task, many approaches have been proposed [37, 16, 39, 15, 5, 31, 8, 36, 6, 9, 32]. As shown in the left of Fig. 1, a VFR model generally consists of two important parts: an image-level feature extractor to get the face descriptor (*e.g.* deep or hand-crafted feature) of each face image and a video modeling module to aggregate face descriptors within a video into a compact video representation. Lots of prior work focus on the latter [37, 16, 39, 15, 5, 31, 8, 36, 6], whereas, few efforts are made in the former, except [9, 32]. In this paper, we focus on the former (image-level feature extractor) based on convolutional neural networks (CNNs). Since subjects in videos are often in movement, video faces suffer from large variations (*e.g.* expression, lighting or pose variations), and they also have video-type noises such as motion blur, out-of-focus blur and low resolution. Therefore, to devise an image-level feature extractor, two intuitions should be kept in mind: (1) the network should be qualified for handling large variations, and (2) the extracted face descriptor should be robust against both video-type noises and large face variations.

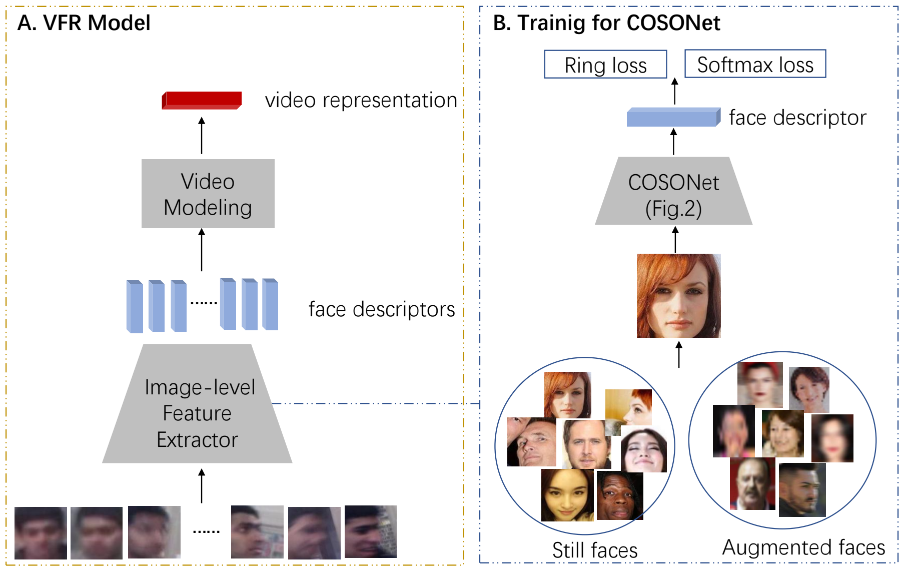

**Fig. 1. Left** is the general VFR model composed of two parts: image-level feature extractor and video modeling module. **Right** is the pipeline of training COSONet. Each training image in a batch is either picked from still faces or video-like augmented ones. The COSONet is trained with two losses: softmax loss and ring loss.

To address the problem of large face variations, CNNs used for image-level feature extractors have been developed into deeper or wider. CASIANet [40](10 layers), VGGFace [27](16 layers), ResNet [35](64 layers), and GoogleNet [39](wider) are typical network structures among them. These networks are all based on first-order feature statistics while rarely consider second-order or higher-order feature statistics. Recently, networks based on second-order feature have gained impressive performance in numerous vision tasks [23, 22, 21, 20, 18]. The second-order feature (*e.g.* sample covariance matrix) encodes the correlation of local features in a translationally invariant way, which is useful to model the texture and appearance of images or regions [34]. In light of the strong modeling capacity and promising performance of the second-order based networks, we aim to obtain a superior network to extract image-level face descriptor based on this technique. After local feature extraction, we perform second-order pooling to compute the sample covariance matrix as the second-order feature.

The sample covariance matrix is a structured representation which has two distinctive attributes: (1) it is a Symmetric Positive Definite (SPD) matrix lying on a specific Riemannian manifold, and (2) it is a high-dimensional matrix (size of $d \times d$ if local features are $d$-dimension). For the first aspect, it is shown that normalizing the sample covariance matrix is essential to achieve better performance [18, 22, 20]. We resort to a stable and fast matrix operation: approximate matrix square root [20] to normalize it. Due to the second attribute, it's necessary to get a compact representation. A straightforward scheme is to flatten the sample covariance matrix into a 1D vector and transform it into lower-dimensional vector by an FC layer. However, there will be plenty of parameters. Instead, we extend the traditional FC layer applied for 1D vectors into 2D matrices, similarly in 2D-PCA [38] and [14, 10], then propose a layer dubbed 2D fully connected (2D-FC) layer with few parameters to obtain a lower-dimensional matrix as the final second-order feature.

Fig. 2 describes the overall network structure, which is composed of four blocks: convolution layers for local feature extraction, second-order pooling for sample covariance matrix estimation, approximate matrix square root for normalization and 2D-FC for a compact representation. We name the network as COmpact Second-Order network (COSONet).

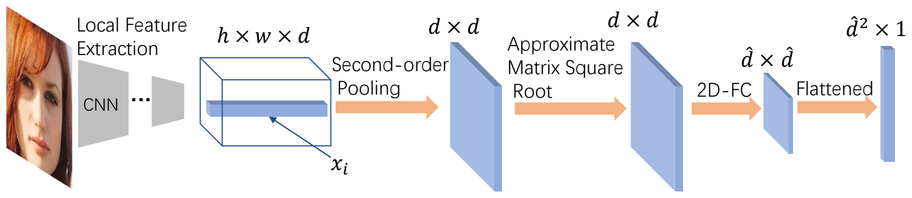

**Fig. 2.** The network structure of COSONet.

For training the COSONet robust against large face variations and video-type noises, face images in training dataset should possess these two types of characteristics. Large still face datasets have large facial variations, but lack of video-type noises. Though large video face datasets have some video-type noises, their face images are redundant and their actual scales are smaller than those still face datasets. Considering the drawbacks of these two types of datasets, we augment video-like faces from large still face dataset by degenerating still faces with video-type noises. Such augmented dataset will be characterized with both large face variations and video-type noises. On the other hand, the dataset will cover face images with varying quality. However, it is found that the $L_2$-norm of low-quality face images' features tends to be smaller than that of high-quality face images', when the widely used softmax loss is the training loss function [29]. As a result, the network is likely to be biased to high-quality images and fails to deal with low-quality ones. To handle such problem, we design our final loss with two terms, where one is the typical softmax loss for encouraging discriminative face descriptors, and the other is the newly developed ring loss [42] for regularizing all face descriptors with equal $L_2$-norm regardless of image qualities.

Right part of Fig. 1 shows our training pipeline. In summary, this paper contributes to the following two aspects:

– For the neural network, we propose COSONet to get a compact second-order image-level face descriptor, which performs better than first-order networks.
– For a well-trained COSONet, we augment video-like face images from large still face datasets and a mixtured loss is used to train the network.

## 2 Related Work

Our method mainly covers two aspects: video face recognition and second-order networks. We will briefly review the works related to these two aspects.

**Video Face Recognition.** Existing video face recognition methods can be roughly divided into two classes. One type of them tries to model facial variations in a video. The other type of them attempts to tackle the low-quality problem in video frames. Approaches of the first type treat frames in a video as an orderless image set and represent the set structure by a variational model which includes affine/convex hull [5], linear subspace [33, 13, 15], statistical models [37, 25, 16] and non-linear manifold [6, 36]. Then, based on the properties of the variational model, a specific metric is induced to compare the similarity of different image sets. This type of methods may be limited in practical usage, because there should be enough frames and facial variations in a video. As most video frames are in low quality due to existing video-type noises, methods of the second type focus on addressing these noises. For instance, [30] generated more discriminative images from multiple degenerating video frames. [39, 24] adopted attention-based pooling to filter out low-quality frames. They first estimated the quality of each frame as its weight and then weakened the effect of low-quality frames by weighted average pooling. In case that all frames in a video are of poor quality, quality scores will be all equally low. As a result, the attention-based pooling will degenerate into average pooling. However, the low-quality problem is not yet solved. Instead, [32, 9] extracted discriminative feature from video frames regardless of their qualities.

These two types of methods all need an image-level feature extractor to get the feature of each frame. In this work, we propose an elaborate network based on second-order networks to fulfill this.

**Second-order Networks.** Second-order networks recently have gained more and more attention. These networks mainly perform second-order pooling after convolution layers. For example, a pioneer work [23] proposed bilinear pooling to model pairwise feature interaction for fine-grained classification and found normalization was essential to improve the performance. Then several normalization methods were proposed. One such method is the matrix-logarithm operation. It maps the second-order representation (*e.g.* the sample covariance matrices) from Symmetric Positive Definite (SPD) matrices manifold to Euclidean space [18]. Later, both [22] and [21] found that matrix power normalization especially matrix square root was more stable and robust than matrix-logarithm. [20] and [22] further accelerated matrix square root by approximate solution. Since the second-order features are high dimensional, [11] adopted two low-dimensional kernel approximation methods to mitigate this problem.

## 3 Approach

In this section, we first introduce our COSONet structure. Then we describe the details of data augmentation and loss functions.

### 3.1 COSONet structure

The overview of our COSONet is depicted in Fig. 2. For a face image, after convolution layers for local feature extraction, we perform second-order pooling to compute the sample covariance matrix of local features, then the sample covariance matrix is normalized by approximate matrix square root. 2D-FC is further applied to get a compact second order representation. Finally, the 2D matrix is flattened into a vector as the face descriptor. Next, we will introduce each block in details.

**Second-order pooling.** Networks that encode second-order statistics information for local features have shown promising results on various vision tasks including fine-grained classification [23, 22], large-scale visual recognition [21, 20], semantic segmentation [18] and face recognition [7]. We exploit it as our image-level feature extractor. Given a face image, let the output of the last convolution layer to be a $h \times w \times d$ tensor with spatial height $h$, spatial width $w$, channel $d$. We reshape it into a matrix $\mathbf{X}$ with size of $n \times d$, where $n = h \times w$, each row of $\mathbf{X}$ is a $d$-dimension local feature $\mathbf{x_i}$. We then perform second-order pooling by estimating the sample covariance matrix as

$$\mathbf{C} = \frac{1}{n-1}\sum_{i=1}^{n}(\mathbf{x_i} - \bar{\mathbf{x}})(\mathbf{x_i} - \bar{\mathbf{x}})^\top \tag{1}$$

where $\mathbf{x_i}$ represents the local feature across location $i = 1, 2, \dots n$. $\bar{\mathbf{x}}$ represents the mean of local features, which is given by $\bar{\mathbf{x}} = \frac{1}{n}\sum_{i=1}^{n}\mathbf{x_i}$. The diagonal elements of $\mathbf{C}$ represent the variance of each feature channel, the off-diagonal elements represent the correlation between different feature channels. The variance or correlation statistics information is useful to model global texture and appearance of a face image.

**Approximate matrix square root normalization.** The sample covariance matrix $\mathbf{C}$ is an SPD matrix lying on a specific Riemannian manifold, directly operating on it is non-trivial. Moreover, to achieve a good performance, it's necessary to normalize it [22, 21, 18]. Currently, there are two normalization strategies: 1) Matrix-logarithm is applied to map the SPD matrices from Riemannian manifold to Euclidean space, specifically, through $\log(\mathbf{C}) = \mathbf{U}\log(\mathbf{S})\mathbf{U}^\top$ with $\mathbf{C} = \mathbf{U}\mathbf{S}\mathbf{U}^\top$ (the SVD for $\mathbf{C}$). 2) Matrix power normalization, especially matrix square root, concretely, through $\mathbf{C}^{\frac{1}{2}} = \mathbf{U}\mathbf{S}^{\frac{1}{2}}\mathbf{U}^\top$. As stated in [21], a more robust sample covariance matrix could be estimated by matrix square root and this normalization was shown to have better performance than matrix logarithm in [22, 21]. Thus it's better to choose matrix square root for normalization. However, matrix square root requires SVD which is numerical unstable during gradient back propagation, and is also time-consuming as it is usually difficult to be accelerated by GPUs.

Instead of accurately calculating matrix square root, we try to find the solution of equation $\boldsymbol{F}(\mathbf{\Sigma}) = \mathbf{\Sigma}^2 - \mathbf{C} = 0$, as advocated in [20]. This equation can be solved by Newton-Schulz iteration efficiently. Specifically, given initial states: $\mathbf{Y}_0 = \mathbf{C}$ and $\mathbf{Z}_0 = \mathbf{I}$, where $\mathbf{I}$ is the identity matrix, the iterative update rule is:

$$\mathbf{Y}_k = \frac{1}{2}\mathbf{Y}_{k-1}(3\mathbf{I} - \mathbf{Z}_{k-1}\mathbf{Y}_{k-1}), \mathbf{Z}_k = \frac{1}{2}(3\mathbf{I} - \mathbf{Z}_{k-1}\mathbf{Y}_{k-1})\mathbf{Z}_{k-1} \tag{2}$$

SVD isn't required, instead, only matrix product is involved which is stable and fast on GPUs. Since the matrices $\mathbf{Y}_k$ can converge to $\mathbf{C}^{\frac{1}{2}}$ quadratically only if $\mathbf{C}$ is in the region of $||\mathbf{C} - \mathbf{I}|| < 1$, to transform any $\mathbf{C}$ into the convergence region, we should first pre-normalize it. Thus, the final approximate matrix square root normalization has three steps:

1. Pre-normalization by the trace as:
$$\hat{\mathbf{C}} = \frac{1}{tr(\mathbf{C})}\mathbf{C} \tag{3}$$
2. Newton-Schulz iteration with $N$ times (a hyper-parameter) through Eq. 2, where the initial states accordingly are $\mathbf{Y}_0 = \hat{\mathbf{C}}$ and $\mathbf{Z}_0 = \mathbf{I}$.
3. Post-normalization to recover the magnitudes and get the matrix square root of $\mathbf{C}$ as:
$$\mathbf{\Sigma} = \sqrt{tr(\mathbf{C})}\mathbf{Y}_N \tag{4}$$

**2D fully connected layer** After normalization, the second-order representation $\mathbf{\Sigma}$ is a matrix with size of $d \times d$. If it is flattened into a vector as the face descriptor and directly sent into the last FC layer of softmax loss function while training, there will be $d \times d \times k$ parameters, where $k$ is the number of classes in the training set. However in large scale face datasets, $k$ may be up to 10,000 (*e.g.* 10,575 subjects in WebFace [40]), if $d$ is 256, there will be 655M parameters! Besides, each face descriptor will be a $d^2$-dimension vector which is too large to store. Considering these problems, it is necessary to get a compact representation. A naive way is to first flatten $\mathbf{\Sigma}$ into a $d^2$-dimension vector and transform it into a lower-dimensional one by traditional FC layer. On the contrary, we apply a more efficient way. We treat $\mathbf{\Sigma}$ as a matrix and extend traditional FC layer applied for 1D vectors into 2D matrice, similarly in [14, 10]. Our specific transformation is given by:

$$\mathbf{H} = \mathbf{W}^{\top}\mathbf{\Sigma}\mathbf{W} \tag{5}$$

where $\mathbf{W} \in \mathbb{R}^{d \times \widehat{d}}$ is a learnable parameter matrix and $\widehat{d}$ is much smaller than $d$. We name this transformation as 2D fully connected layer (2D-FC) as in [10]. After 2D-FC, $\mathbf{H}$ is flattened into a vector ($\mathbf{f} = \mathbf{H}(:)$) to get a $\widehat{d}^2$-dimension face descriptor. It can be seen that the 2D-FC layer has much fewer parameters, compared to the naive way (FC layer operated on the flattened $\mathbf{\Sigma}(:)$). FC layer would have $d^2 \times \widehat{d}^2$ parameters to get the same dimensional face descriptor, whereas, 2D-FC layer merely needs $d \times \widehat{d}$ parameters.

## 3.2 Data augmentation

For COSONet automatically extracting robust feature against large face variations and video-type noises, the training datasets should have these two types of data characteristics. Large still face datasets such as WebFace [40] and VGGFace2 [4] have large face variations but no video-type noises. Large video face datasets such as UMDFaces-Video [1] have video-type noises, but face images contain some redundancy because nearby faces are similar for some videos. The actual scale of video face datasets is smaller than those still face datasets with similar number of face images. Considering the shortcoming of these two kinds of existing datasets, we propose to augment video-like faces from the given large still face dataset for training. Such augmented dataset will be characterized with both large face variations and video-type noises. Concretely, we apply three types of augmentation strategies as similarly done in [9, 32]:

- Motion blur: we generate motion blur-like images with linear kernels whose length and angle are randomly chosen from {11, 15, 19} and {0, 45, 90, 135} respectively.
- Out-of-focus blur: we simulate out-of-focus blur by Gaussian kernels whose size and width are randomly chosen from {9, 11, 15} and {1.5, 3.0, 4.5}.
- Low resolution: we resize images with scales randomly chosen from $\{\frac{1}{2}, \frac{1}{4}, \frac{1}{6}\}$.

Three types of transformations are employed sequentially into a face image with probability of 0.5. The first two strategies are inspired by the generally explored principles in deblurring works where the blur effects can be modeled as some convolution operations between the original image and blur filter kernels. Fig. 3 shows some augmented images. The COSONet is trained on the mixtured face datasets containing still faces and augmented faces with video-type noises.

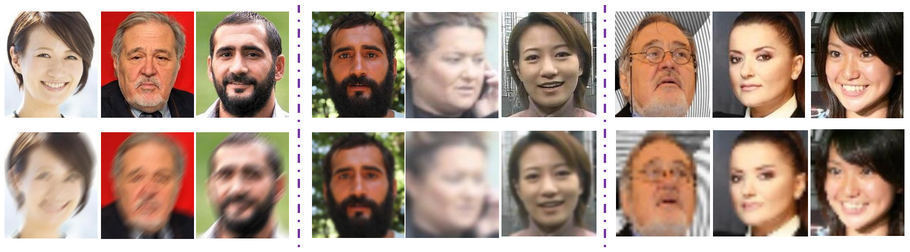

**Fig. 3.** Augmented video-like images. Upper row is the original still faces. Left, middle and right section of the lower row are augmented faces with motion blur, out-of-focus blur and low resolution respectively.

### 3.3 Loss function

We use softmax loss function to train our COSONet, as it can converge much fast, and at the same time it can get impressive performance. But another problem should be also noticed. After data augmentation, face images in the dataset are in various quality. As noted in [29], features for low-quality face images tend to have smaller $L_2$-norm comparted to high-quality ones if only softmax loss is used. In such situation, low-quality face images are more likely to be ignored while training. To alleviate this issue, we should balance features' $L_2$-norm regardless of image qualities. Specifically, we utilize the lately proposed ring loss [42] to regularize the features, which is given by:

$$L_r = (\|\mathbf{f}\|_2 - R)^2 \tag{6}$$

where $R$ is a learnable scale parameter. Through ring loss, the network can learn to normalize the feature on a hypersphere with the same $L_2$-norm. Finally, our loss function becomes:

$$L_s = -\log \frac{e^{\mathbf{W}_k^\top \mathbf{f} + \mathbf{b}_k}}{\sum_{i=1}^{K} e^{\mathbf{W}_i^\top \mathbf{f} + \mathbf{b}_i}} + \lambda L_r \tag{7}$$

where $k$ is the ground-truth label and $K$ is the total number of subjects. $L_s$ contains two terms, the former is softmax loss to get discriminative face descriptors, and the latter is ring loss to constrain the $L_2$-norm of face descriptors. A scalar $\lambda$ is used for balancing the two loss functions.

## 4 Experiments

We divide our experiments into several sections. In the first section, we introduce our datasets and evaluation protocols. In the second section, we present our implementation details. In the third section, we conduct component analysis for each technique of the proposed method. In the fourth section, we compare with state-of-the-art methods. Then we visualize what are learned by the networks. Finally, we provide a further discussion.

### 4.1 Datasets and evaluation protocols

We have two training datasets, one is WebFace [40] with about 0.5M images of 10,575 subjects, and the other one is the recent published VGGFace2 [4] with 3.1M images of 8,631 subjects. Since WebFace is small, we train on it for our later detailed component analysis (in Sec. 4.3). VGGFace2 is quite large, we train on it to get comparable performance with the state-of-the art methods (in Sec. 4.4). We evaluate our method on two datasets: IARPA Janus Benchmark A (IJB-A) [19] and Point-and-Shoot Challenge (PaSC) [2].

IJB-A dataset contains 5,712 still images and 20,414 video frames of 500 subjects. All images and frames are captured in unconstrained conditions and cover large variations in pose and image quality. Each sample or instance in IJB-A is called a 'template' which consists of the mixture between still images and video frames. IJB-A provides two protocols: 1:1 face verification and 1: N face identification. We only focus on the former protocols where 10-fold testing is conducted. PaSC dataset includes 2,802 videos of 265 subjects. Half of its videos are captured by hand held camera (denoted as PaSC-H), the rest are captured by controlled camera (denoted as PaSC-C). Subjects in the dataset are asked to do predesigned actions. Thus, faces cover large pose variations and serious video-type noises. We test on PaSC dataset with the provided face verification protocol.

We report the true acceptance rate (TAR) at different false acceptance rates (FARs).

### 4.2 Implementation details

**Preprocessing.** We use MTCNN [41] algorithm to detect faces in both training and testing datasets. The bounding box is extended by a factor of 0.3 to include some context. The shorter side of each image is resized into 256 and the other side is resized accordingly to keep the original ratio. While training, a region of $224 \times 224$ is randomly cropped from an image or its horizontal flip, with the per-pixel mean subtracted. While testing, a region of $224 \times 224$ is cropped from the center of each image.
**Detailed network configurations.** We implement our method with PyTorch [3] [28]. The local feature extraction block of our COSONet is based on two ResNet-type networks: ResNet-18 and ResNet-34. We slightly change the CNN structure to obtain 196 local features after the last convolution layers. ResNet-type networks have five convolution blocks. We keep the first 4 convolution blocks the same. A convolution layer for down sampling in the 5th convolution block is canceled. The details of network structure are given in Table 1. We empirically run Newton-Schulz iteration 5 times (N = 5), following [20] and set the output size of 2D-FC to be $64 \times 64$. Compared with ResNet, COSONet only brings extra $256 \times 64$ parameters in 2D-FC layer ($\mathbf{W}$ in Eq. 5). More detailed settings are provided in the supplemental material.

[3] The source code is available at *http://vipl.ict.ac.cn/resources/codes.*

**Table 1.** The network structure of plain ResNet-type networks and our COSONet structure. SO_normed: second order pooling and approximate matrix square root normalization. conv_co: $1 \times 1 \times 256$ convolution layer for compressing the channels.

| **Plain ResNet-18 or ResNet-34** | | | | | | |
|---|---|---|---|---|---|---|
| layer name | conv(1-4) | conv5 | average pooling | | | Flattened |
| output size | 14×14×256 | 7×7×512 | 1×1×512 | | | 512-d |
| **Our COSONet** | | | | | | |
| layer name | conv(1-4) | conv5 | conv_co | SO_normed | 2D-FC | Flattened |
| output size | 14×14×256 | 14×14×512 | 14×14×256 | 256×256 | 64×64 | 4096-d |

**Detailed settings in testing.** For face images in a template (in IJB-A) or video (in PaSC), each face image is forwarded into the network to get its face descriptor. Since we focus on the image-level feature extractors rather than video modeling modules, we just apply simple video aggregation method, average pooling across face descriptors in the template or video to get a compact video representation. Finally, the similarity between two video representations is computed by the cosine similarity.

## 4.3 Component analysis

**Effect of the COSONet structure.** In this part, we mainly study the effect of COSONet structure. We implement our COSONet based on two ResNet-type networks: ResNet-18 and ResNet-34 as mentioned above. These two networks and their COSONet are trained from scratch on WebFace. The initial learning rate is 0.2 and decreased by a factor of 0.5 for every 3 epochs. The batch size is 256. For fair comparison, we only use softmax loss as the loss function. Table 2 shows that our COSONet can improve the performance by a large margin no matter what type of networks is used. It indicates that our COSONet has stronger modeling capacity than plain fisrt-order networks to deal with large face variations in videos.

**Table 2.** Performance comparison for plain network and COSONet on IJB-A and PaSC. '–' and '✓' represent plain network and COSONet respectively. $10^{-3}$ and $10^{-2}$ indicate different FARs.

| Method | | IJB-A | | PaSC-H | | PaSC-C | |
|---|---|---|---|---|---|---|---|
| CNN | COSO | $10^{-3}$ | $10^{-2}$ | $10^{-3}$ | $10^{-2}$ | $10^{-3}$ | $10^{-2}$ |
| ResNet-18 | – | 0.567±0.058 | 0.788±0.025 | 0.029 | 0.266 | 0.352 | 0.749 |
| ResNet-18 | ✓ | **0.616±0.043** | **0.811±0.024** | **0.088** | **0.400** | **0.613** | **0.873** |
| ResNet-34 | – | 0.591±0.056 | 0.813±0.025 | 0.080 | 0.381 | 0.506 | 0.844 |
| ResNet-34 | ✓ | **0.682±0.045** | **0.858±0.019** | **0.185** | **0.581** | **0.730** | **0.927** |

**Table 3.** Performance comparison for without or with effect of data augmentation on IJB-A and PaSC. '–' and '✓' represent without and with data augmentation respectively. $10^{-3}$ and $10^{-2}$ indicate different FARs. The ROC curve is shown in supplemental material.

| Method | | IJB-A | | PaSC-H | | PaSC-C | |
|---|---|---|---|---|---|---|---|
| Model | Aug | $10^{-3}$ | $10^{-2}$ | $10^{-3}$ | $10^{-2}$ | $10^{-3}$ | $10^{-2}$ |
| A | – | 0.591±0.056 | 0.813±0.025 | 0.080 | 0.381 | 0.506 | 0.844 |
| B | ✓ | **0.644±0.057** | **0.841±0.019** | **0.182** | **0.478** | **0.536** | **0.845** |

**Effect of data augmentation.** In this part, we fix the network structure as ResNet-34 and loss function as softmax loss to study the effect of data augmentation. For WebFace (0.5M), we augment the same number of images with the original data. ResNet-34 is first trained on WebFace dataset (0.5M) and then is finetuned on the augmented WebFace dataset (total 1.0M). Table 3 presents the comparison results. Notably, the base network trained on WebFace performs badly on PaSC hand held mode but after it is finetuned on the augmented WebFace, they can get much improvement. This is mainly because faces in PaSC hand held mode are in serious capture condition and have a large domain gap with still faces in WebFace. Through augmentation, the gap could be narrowed. On other hand, compared to the hand held mode, the improvement on controlled mode is slightly small, which demonstrates that faces in controlled mode are similar to still faces. Overall, data augmentation can improve the performance. We attribute this to the fact that the network can automatically learn robust feature against video-type noises from the augmented data.

**Effect of ring loss for our proposed method.** As the effectiveness of ring loss has been validated in [42] with only first-order CNNs, in this part, we verify its influence for our COSONet and data augmentation strategy. The local feature extraction block of COSONet is based on ResNet-34. Training data are the same as the last part. Table 4 displays the result. For Model I (COSONet only with softmax loss) and Model II (COSONet with softmax loss and ring loss), we can see that ring loss also takes effects on our COSONet. For Model II and Model

**Table 4.** Performance comparison for the impact of ring loss. '–' and '✓' represent without or with the corresponding technique. The ROC curve is shown in supplemental material.

| | | | | IJB-A | | PaSC-H | | PaSC-C | |
|---|---|---|---|---|---|---|---|---|---|
| Model | COSO | Aug | ring loss | $10^{-3}$ | $10^{-2}$ | $10^{-3}$ | $10^{-2}$ | $10^{-3}$ | $10^{-2}$ |
| I | ✓ | – | – | 0.682±0.045 | 0.858±0.019 | 0.185 | 0.581 | 0.730 | 0.927 |
| II | ✓ | – | ✓ | 0.735±0.042 | 0.885±0.014 | 0.320 | 0.693 | 0.794 | 0.935 |
| III | ✓ | ✓ | ✓ | 0.758±0.037 | 0.894±0.014 | 0.470 | 0.805 | 0.829 | 0.951 |

**Table 5.** Comparative performance on IJB-A dataset. The network of model D$^\dagger$ and D$^{'}$ are the same as model D, but their pooling schemes are different in test stage. D$^\dagger$ utilizes media-pooling where images in each media are first aggregated separately by average pooling according to the media id, as in [39]. D$^{'}$ adopts the SoftMax score pooling which is also applied in [26, 42]. $^*$: Ring loss [42] didn't provide the result at FAR=$10^{-2}$ and the standard deviation in its paper.

| Method | | | | $10^{-3}$ | $10^{-2}$ |
|---|---|---|---|---|---|
| **Existing Methods** | | | | | |
| Ring loss [42] | | | | 0.915$^*$ | – |
| VGGFace2 [4] | | | | **0.921±0.014** | **0.968±0.006** |
| Quality Aware Network [24] | | | | 0.893±0.392 | 0.942±0.153 |
| Neural Aggregation Network [39] | | | | 0.881±0.011 | 0.941±0.008 |
| DREAM [3] | | | | 0.868±0.015 | 0.944±0.009 |
| Template adaptation [8] | | | | 0.836±0.027 | 0.939±0.013 |
| Data augmentation + Video Pooling [26] | | | | 0.725±0.044 | 0.886±0.017 |
| Unsupervised Domain Adaptation [32] | | | | 0.649±0.022 | 0.864±0.007 |
| **Our approach** | | | | | |
| Model | COSO | Aug | ring loss | | |
| A | – | – | – | 0.854±0.032 | 0.949±0.010 |
| B | ✓ | – | – | 0.881±0.023 | 0.957±0.006 |
| C | ✓ | ✓ | – | 0.883±0.024 | 0.952±0.006 |
| D | ✓ | ✓ | ✓ | 0.902±0.022 | 0.958±0.005 |
| D$^\dagger$ | ✓ | ✓ | ✓ | 0.915±0.015 | 0.962±0.005 |
| D$^{'}$ | ✓ | ✓ | ✓ | 0.913±0.014 | 0.963±0.004 |

III (with data augmentation), data augmentation gets similar improvement as the last part.

## 4.4 Comparison with state-of-the-art methods

In this section, we compare our whole approach with the state-of-the-art methods. To get a more powerful COSONet, we train the network on VGGFace2 and its augmented data where we augment 1.0M video-like images. For the local feature extraction of COSONet, we implement it based on ResNet-34.

Table 5 and Table 6 present our best performance on IJB-A and PaSC, as well as previously reported results by state-of-the-art methods. We also gradually add each technique to evaluate its influence. The results of other works are directly copied from their original papers.

First, in view of Model A and Model B, Model A is trained with softmax loss as a baseline. When we change the network to our COSONet (Model B), we observe consistent improvement. On IJB-A dataset, for example, TAR@FAR=$10^{-3}$ increases from 0.854 to 0.881 and even a larger gain on PaSC hand held mode, from 0.429 to 0.596.

Second, for Model B and Model C, we finetune Model C from Model B on the augmented dataset with our generated 1.0M video-like images. Although there

**Table 6.** Comparative performance on PaSC dataset.

| Method | | | | PaSC-H | | PaSC-C | |
|---|---|---|---|---|---|---|---|
| | | | | $10^{-3}$ | $10^{-2}$ | $10^{-3}$ | $10^{-2}$ |
| **Existing Methods** | | | | | | | |
| Trunk-Branch Ensemble CNN [9] | | | | – | **0.962** | – | 0.958 |
| Attention-aware [31] | | | | – | 0.937 | – | 0.956 |
| DAN [30] | | | | – | 0.803 | – | 0.920 |
| CERML [16] | | | | – | 0.773 | – | 0.801 |
| SPDNet [14] | | | | – | 0.728 | – | 0.801 |
| GrassmannNet [17] | | | | – | 0.727 | – | 0.805 |
| **Our approach** | | | | | | | |
| Model | COSO | Aug | ring loss | | | | |
| A | – | – | – | 0.429 | 0.811 | 0.858 | 0.955 |
| B | ✓ | – | – | 0.596 | 0.879 | 0.884 | 0.962 |
| C | ✓ | ✓ | – | 0.765 | 0.934 | 0.902 | 0.966 |
| D | ✓ | ✓ | ✓ | 0.852 | 0.960 | 0.927 | **0.974** |

is no improvement on IJB-A, on PaSC hand held mode the improvement is prominent, from 0.596 to 0.765 when TAR@FAR=$10^{-3}$. This is mainly because faces in PaSC hand held mode are in serious captured condition as mentioned earlier.

Finally, we finetune Model D from Model C with softmax loss and ring loss to get the ultimate model. On IJB-A dataset, we apply different pooling schemes other than the average feature pooling, during testing. Model $D^{\dagger}$ and Model $D^{'}$ explore media pooling [39] and SoftMax score pooling [26] respectively. We get comparable performance against other works. Ring loss [42] utilized the same loss function and the same pooling scheme (SoftMax score pooling) in test stage with our Model $D^{'}$. Its network (ResNet-64) is deeper than ours but the performance is similar. The network of VGGFace2 [4] is trained on VGGFace2 dataset and then finetuned on MS-Celeb-1M [12]. Their datasets are larger than ours but the performance is also similar. On PaSC dataset, we get comparable and even better performance than other works on hand held and controlled mode respectively. The method of Trunk-Branch Ensemble CNN [9] finetuned their network on PaSC, whereas, we didn't.

## 4.5 Feature Visualization

To further investigate the effectiveness of our COSONet, we compare its feature response with the plain first-order network in Fig. 4. It can be seen that our COSONet can automatically focus on discriminative face regions, which has illustrated its effective modeling capacity to mitigate the pose variation problem.

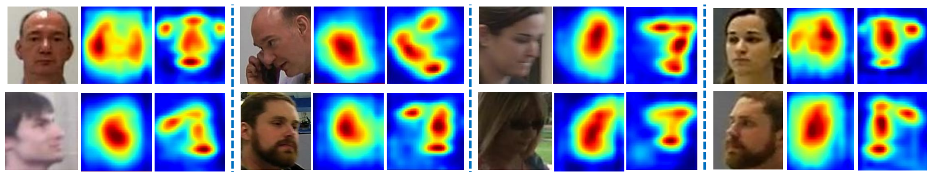

**Fig. 4.** In each group, the first is the inputed face image, the second and the third are the corresponding averaged feature maps along the channel dimension of the plain first-order network and our COSONet. The two feature maps are respectively from the output of the fourth block convolution layer of Model A and Model B (in Sec. 4.4.).

### 4.6 Discussion

As for media pooling on IJB-A dataset (Model $D^{\dagger}$ in Table 5), there are two types of media images in a template, video frames and still images. As [39], we first aggregate images in each media by average pooling and then combine these two media features by averaging. The exact equation is given by

$$\mathbf{v} = \frac{1}{2|\mathcal{S}|}\sum_{\mathbf{f}\in\mathcal{S}}\mathbf{f} + \frac{1}{2|\mathcal{V}|}\sum_{\mathbf{f}\in\mathcal{V}}\mathbf{f} \tag{8}$$

where $\mathcal{S}$ and $\mathcal{V}$ represent the set of face descriptors for still images and video frames within a template, respectively. $|\cdot|$ is the size of a set. $\mathbf{v}$ is the video representation. We observe media pooling can get some enhancement in our paper and other papers [39, 8]. We dig slightly deeper into this. We find the average ratio between the number of video frames ($|\mathcal{V}|$) and the number of still images ($|\mathcal{S}|$) in each template is $7.36 : 1$. This means that media pooling is actually a weighted average pooling. Weights for video frames are smaller than those for still images. However, this pooling scheme is biased to IJB-A dataset.

## 5 Conclusions

We propose a whole framework for extracting robust image-level face descriptors for VFR. Extensive experiments have validated the superiority of each technique and the proposed method. Existing video modeling methods can be integrated into the framework to further refine the video representation. Nevertheless, we attribute the good performance of our method to three aspects: 1) COSONet is superior to first-order network to handle large face variations. In future we will fuse multiple order information to encode local features. 2) data augmentation can upgrade the robustness of the face descriptors, and 3) an elaborate mixture loss function is adopted to the data characteristics.

**Acknowledgements.** This work is partially supported by Natural Science Foundation of China under contracts Nos. 61390511, 61772500, 973 Program under contract No. 2015CB351802, Frontier Science Key Research Project CAS No. QYZDJ-SSW-JSC009, and Youth Innovation Promotion Association CAS No. 2015085.

## References

1. Bansal, A., Castillo, C., Ranjan, R., Chellappa, R.: The Do's and Don'ts for CNN-Based face verification. In: ICCV Workshop. pp. 2545–2554 (2017)
2. Beveridge, J.R., Phillips, P.J., Bolme, D.S., Draper, B.A.: The challenge of face recognition from digital point-and-shoot cameras. In: ICB. pp. 1–8 (2013)
3. Cao, K., Rong, Y., Li, C., Tang, X., Loy, C.C.: Pose-robust face recognition via deep residual equivariant mapping. In: CVPR. pp. 5187–5196 (2018)
4. Cao, Q., Shen, L., Xie, W., Parkhi, O.M., Zisserman, A.: VGGFace2: A dataset for recognising faces across pose and age. In: arXiv:1710.08092 (2017)
5. Cevikalp, H., Triggs, B.: Face recognition based on image sets. In: CVPR. pp. 2567–2573 (2010)
6. Chen, S., Sanderson, C., Harandi, M.T., Lovell, B.C.: Improved image set classification via joint sparse approximated nearest subspaces. In: CVPR. pp. 452–459 (2013)
7. Chowdhury, A.R., Lin, T.Y., Maji, S., Learnedmiller, E.: One-to-many face recognition with bilinear cnns. In: WACV. pp. 1–9 (2016)
8. Crosswhite, N., Byrne, J., Stauffer, C., Parkhi, O., Cao, Q., Zisserman, A.: Template adaptation for face verification and identification. In: FG. pp. 1–8 (2017)
9. Ding, C., Tao, D.: Trunk-branch ensemble convolutional neural networks for video-based face recognition. In: IEEE TPAMI. pp. 1002–1014 (2018)
10. Dong, Z., Jia, S., Zhang, C., Pei, M., Wu, Y.: Deep manifold learning of symmetric positive definite matrices with application to face recognition. In: AAAI. pp. 4009–4015 (2018)
11. Gao, Y., Beijbom, O., Zhang, N., Darrell, T.: Compact bilinear pooling. In: CVPR. pp. 317–326 (2016)
12. Guo, Y., Zhang, L., Hu, Y., He, X., Gao, J.: MS-Celeb-1M: A dataset and benchmark for large-scale face recognition. In: ECCV. pp. 87–102 (2016)
13. Hamm, J., Lee, D.D.: Grassmann discriminant analysis: a unifying view on subspace-based learning. In: ICML. pp. 376–383 (2008)
14. Huang, Z., Gool, L.V.: A riemannian network for spd matrix learning. In: AAAI. pp. 2036–2042 (2017)
15. Huang, Z., Wang, R., Shan, S., Chen, X.: Projection metric learning on grassmann manifold with application to video based face recognition. In: CVPR. pp. 140–149 (2015)
16. Huang, Z., Wang, R., Shan, S., Gool, L.V., Chen, X.: Cross euclidean-to-riemannian metric learning with application to face recognition from video. In: IEEE TPAMI (2018), https://doi.org/10.1109/TPAMI.2017.2776154
17. Huang, Z., Wu, J., Gool, L.V.: Building deep networks on grassmann manifolds. In: AAAI. pp. 3279–3286 (2018)
18. Ionescu, C., Vantzos, O., Sminchisescu, C.: Matrix backpropagation for deep networks with structured layers. In: ICCV. pp. 2965–2973 (2015)
19. Klare, B.F., Jain, A.K., Klein, B., Taborsky, E., Blanton, A., Cheney, J., Allen, K., Grother, P., Mah, A., Burge, M.: Pushing the frontiers of unconstrained face detection and recognition: Iarpa janus benchmark a. In: CVPR. pp. 1931–1939 (2015)
20. Li, P., Xie, J., Wang, Q., Gao, Z.: Towards faster training of global covariance pooling networks by iterative matrix square root normalization. In: CVPR. pp. 947–955 (2018)

21. Li, P., Xie, J., Wang, Q., Zuo, W.: Is second-order information helpful for large-scale visual recognition? In: ICCV. pp. 2089–2097 (2017)
22. Lin, T.Y., Maji, S.: Improved bilinear pooling with CNNs. In: BMVC (2017), CoRR abs/1707.06772
23. Lin, T.Y., Roychowdhury, A., Maji, S.: Bilinear cnn models for fine-grained visual recognition. In: ICCV. pp. 1449–1457 (2015)
24. Liu, Y., Yan, J., Ouyang, W.: Quality aware network for set to set recognition. In: CVPR. pp. 4694–4703 (2017)
25. Lu, J., Wang, G., Moulin, P.: Image set classification using holistic multiple order statistics features and localized multi-kernel metric learning. In: ICCV. pp. 329–336 (2013)
26. Masi, I., Trn, A.T., Hassner, T., Leksut, J.T., Medioni, G.: Do we really need to collect millions of faces for effective face recognition? In: ECCV. pp. 579–596 (2016)
27. Parkhi, O.M., Vedaldi, A., Zisserman, A.: Deep face recognition. In: BMVC. pp. 1–12 (2015)
28. Paszke, A., Gross, S., Chintala, S., Chanan, G., Yang, E., DeVito, Z., Lin, Z., Desmaison, A., Antiga, L., Lerer, A.: Automatic differentiation in pytorch. In: NIPS Workshop (2017)
29. Ranjan, R., Castillo, C.D., Chellappa, R.: L2-constrained softmax loss for discriminative face verification. In: arXiv:1703.09507 (2017)
30. Rao, Y., Lin, J., Lu, J., Zhou, J.: Learning discriminative aggregation network for video-based face recognition. In: ICCV. pp. 3801–3810 (2017)
31. Rao, Y., Lu, J., Zhou, J.: Attention-aware deep reinforcement learning for video face recognition. In: ICCV. pp. 3951–3960 (2017)
32. Sohn, K., Liu, S., Zhong, G., Yu, X., Yang, M.H., Chandraker, M.: Unsupervised domain adaptation for face recognition in unlabeled videos. In: ICCV. pp. 5917–5925 (2017)
33. Tae-Kyun, K., Josef, K., Roberto, C.: Discriminative learning and recognition of image set classes using canonical correlations. In: IEEE TPAMI. pp. 1005–1018 (2007)
34. Tuzel, O., Porikli, F., Meer, P.: Region covariance: A fast descriptor for detection and classification. In: ECCV. pp. 589–600 (2006)
35. Wang, H., Wang, Y., Zhou, Z., Ji, X., Gong, D., Zhou, J., Li, Z., Liu, W.: CosFace: Large margin cosine loss for deep face recognition. In: CVPR. pp. 5265–5274 (2018)
36. Wang, R., Chen, X.: Manifold discriminant analysis. In: CVPR. pp. 429–436 (2009)
37. Wang, W., Wang, R., Shan, S., Chen, X.: Discriminative covariance oriented representation learning for face recognition with image sets. In: CVPR. pp. 5749–5758 (2017)
38. Yang, J., Zhang, D., Frangi, A.F., Yang, J.Y.: Two-dimensional PCA: a new approach to appearance-based face representation and recognition. In: IEEE TPAMI. pp. 131–137 (2004)
39. Yang, J., Ren, P., Chen, D., Wen, F., Li, H., Hua, G.: Neural aggregation network for video face recognition. In: CVPR. pp. 5216–5225 (2017)
40. Yi, D., Lei, Z., Liao, S., Li, S.Z.: Learning face representation from scratch. In: arXiv:1411.7923 (2014)
41. Zhang, K., Zhang, Z., Li, Z., Qiao, Y.: Joint face detection and alignment using multitask cascaded convolutional networks. In: IEEE SPL. pp. 1499–1503 (2016)
42. Zheng, Y., Pal, D.K., Savvides, M.: Ring loss: Convex feature normalization for face recognition. In: CVPR. pp. 5089–5097 (2018)