Covariance Pooling for Facial Expression Recognition

Dinesh Acharya[†], Zhiwu Huang[†], Danda Pani Paudel[†], Luc Van Gool[†‡]
[†]Computer Vision Lab, ETH Zurich, Switzerland [‡]VISICS, KU Leuven, Belgium
{acharyad, zhiwu.huang, paudel, vangool}@vision.ee.ethz.ch

Abstract

Classifying facial expressions into different categories requires capturing regional distortions of facial landmarks. We believe that second-order statistics such as covariance is better able to capture such distortions in regional facial features. In this work, we explore the benefits of using a manifold network structure for covariance pooling to improve facial expression recognition. In particular, we first employ such kind of manifold networks in conjunction with traditional convolutional networks for spatial pooling within individual image feature maps in an end-to-end deep learning manner. By doing so, we are able to achieve a recognition accuracy of 58.14% *on the validation set of Static Facial Expressions in the Wild (SFEW 2.0) and* 87.0% *on the validation set of Real-World Affective Faces (RAF) Database*[1]*. Both of these results are the best results we are aware of. Besides, we leverage covariance pooling to capture the temporal evolution of per-frame features for video-based facial expression recognition. Our reported results demonstrate the advantage of pooling image-set features temporally by stacking the designed manifold network of covariance pooling on top of convolutional network layers.*

1. Introduction

Facial expressions play an important role in communicating the state of our mind. Both humans and computer algorithms can greatly benefit from being able to classify facial expressions. Possible applications of automatic facial expression recognition include better transcription of videos, movie or advertisement recommendations, detection of pain in telemedicine etc.

Traditional convolutional neural networks (CNNs) that use convolutional layers, max or average pooling and fully connected layers are considered to capture only first-order statistics [25]. Second-order statistics such as covariance are considered to be better regional descriptors than first-order statistics such as mean or maximum [20]. As shown in Figure 1, facial expression recognition is more directly related to how facial landmarks are distorted rather than presence or absence of specific landmarks. We believe that second-order statistics is more suited to capture such distortions than first-order statistics. To learn second-order information deeply, we introduce covariance pooling after final convolutional layers. For further dimensionality reduction we borrow the concepts from the manifold network [11] and train it together with conventional CNNs in an end-to-end fashion. It is important to point out this is not a first work to introduce second-order pooling to traditional CNNs. Covariance pooling was initially used in [13] for pooling covariance matrix from the outputs of CNNs. [25] proposed an alternative to compute second-order statistics in the setting of CNNs. However, such two works do not use either dimensionality reduction layers or non-linear rectification layers for second-order statistics. In this paper, we present a strong motivation for exploring them in the context of facial expression recognition.

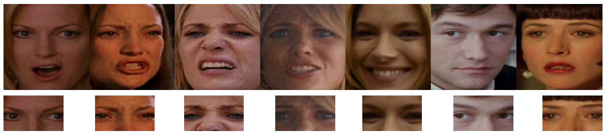


Figure 1. Top: sample images of different facial expression classes from the SFEW dataset. Bottom: distortion of region between two eyebrows in the corresponding facial images.

In addition to being better able to capture distortions in regional facial features, covariance pooling can also be used to capture temporal evolution of per-frame features. Covariance matrix has been employed before to summarize per-frame features [17]. In this work, we experiment with using manifold networks for pooling per-frame features.

In summary, the contribution of this paper is two-fold:

- End-to-end pooling of second-order statistics for both videos and images in the context of facial expression recognition
- State-of-art result on image-based facial expression recognition

[1] The code of this paper will be eventually released on https://github.com/d-acharya/CovPoolFER

2. Related Works

Though facial expression recognition from both images and videos are closely related, they each have their own challenges. Videos contain dynamic information which a single image lacks. With this additional dynamic information, we should theoretically be able to improve facial expression accuracy. However, extracting information from videos has its own challenges. In following sub-sections, we briefly review standard approaches to facial expressions on both image and video-based approaches.

2.1. Facial Expression Recognition from Images

Most of the recent approaches in facial expression recognition from images use various standard architectures such as VGG networks, Inception networks, Residual networks, Inception-Residual Networks etc [3][7][21]. Many of these works carry out pretraining on FER-2013, face recognition datasets or similar datasets and either use outputs from fully connected layers as features to train classifiers or fine-tune the whole network. Use of ensemble of multiple CNNs and fusion of the predicted scores is also widely used and found to be successful. For example, in Emotiw2015 sub challenge on image-based facial expression recognition, both winners and runner up teams [15][26] employed ensemble of CNNs to achieve the best reported score. There, pretraining was done on FER-2013 dataset. Recently, in [3], authors reported validation accuracy of 54.82% which is a state-of-art result for a single network. The accuracy was achieved using VGG-VD-16. The authors carried out pretraining on VGGFaces and FER-2013.

All such networks discussed above employ traditional neural network layers. These architectures can be considered to capture only first-order statistics. Covariance pooling, on the other hand captures second-order statistics. One of the earliest works employing covariance pooling for feature extraction used it as regional descriptor [6][20]. In [25], authors propose various architectures based on VGG network to employ covariance pooling. In [11], authors present a deep learning architecture for learning on Riemannian manifold which can be employed for covariance pooling.

2.2. Facial Expression Recognition from Videos

Traditionally, video-based recognition problems used per-frame features such as SIFT, dense-SIFT, HOG [17] and recently deep features extracted with CNNs have been used [9] [4]. The per-frame features are then used to assign score to each individual frame. Summary statistics of such per-frame features are then used for facial expression recognition. In [24], authors propose modification of Inception architecture to capture action unit activation which can be beneficial for facial expression recognition. Other works use various techniques to capture the temporal evolution of the per-features. For example, LSTMs have been successfully employed with various names such as CNN-RNN, CNN-BRNN etc [8][9][23]. 3D convolutional neural networks have also been used for facial expression recognition. However, performance of a single 3D-ConvNet was worse than applying LSTMs on per-frame features [9]. State-of-art result reported in [9] was obtained by score fusion of multiple models of 3D-ConvNets and CNN-RNNs.

Covariance matrix representation was used as one of the summary statistics of per-frame features in [17]. Kernel-based partial least squares (PLS) were then used for recognition. Here, we use the methods in [17] as baseline and use the SPD Riemannian networks instead of kernel based PLS for recognition and obtain slight improvement.

3. Facial Expression Recognition and Covariance Pooling

3.1. Overview

Facial expression is localized in the facial region whereas images in the wild contain large irrelevant information. Due to this, face detection is performed first and then aligned based on facial landmark locations. Next, we feed the normalized faces into a deep CNN. To pool the feature maps spatially from the CNN, we propose to use covariance pooling, and then employ the manifold network [11] to deeply learn the second-order statistics. The pipeline of our proposed model for image-based facial expression recognition is shown in Figure 2.

As the case of image-based facial expression recognition, videos in the wild contain large irrelevant information. First, all the frames are extracted from a video. Face detection and alignment is then performed on each individual frame. Depending on the feature extraction algorithm, either image features are extracted from the normalized faces or the normalized faces are concatenated and 3d convolutions are applied to the concatenated frames. Intuitively, as the temporal convariance can capture the useful facial motion pattern, we propose to pool the frames over time. To deeply learn the temporal second-order information, we also employ the manifold network [11] for dimensionality reduction and non-linearity on covariance matrices. The overview of our presented model for video-based facial expression recognition is illustrated in Figure 3.

Accordingly, the core techniques of the two proposed models are spatial/temporal covariance pooling and the manifold network for learning the second-order features deeply. In the following we will introduce the two crucial techniques.

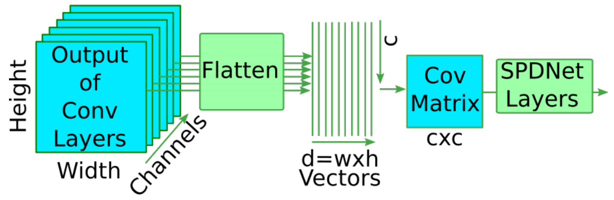


Figure 2. In order to leverage covariance pooling on image-based facial expression recognition problem, output of convolutional layer is flattened as illustrated. The covariance matrix is computed form resulting vectors.

3.2. Covariance Pooling

As discussed earlier, traditional CNNs that consist of fully connected layers, max or average pooling and convolutional layers only capture first-order information [25]. ReLU introduces non-linearity but does so only at individual pixel level. Covariance matrices computed from features are believed to be better able to capture regional features than first-order statistics [20].

Given a set of features, covariance matrix can be used to compactly summarize the second-order information in the set. If $\mathbf{f_1}, \mathbf{f_2}, \ldots, \mathbf{f_n} \in \mathbb{R}^d$ be the set of features, the covariance matrix can be computed as:

$$\mathbf{C} = \frac{1}{n-1} \sum_{i=1}^{n} (\mathbf{f_i} - \bar{\mathbf{f}})(\mathbf{f_i} - \bar{\mathbf{f}})^T, \tag{1}$$

where $\bar{\mathbf{f}} = \frac{1}{n} \sum_{i=1}^{n} \mathbf{f_i}$.

The matrices thus obtained are symmetric positive definite (SPD) only if number of linearly independent components in $\{\mathbf{f_1}, \mathbf{f_2}, \ldots, \mathbf{f_n}\}$ is greater than d. In order to employ the geometric structure preserving layers of the SPD manifold network [11], the covariance matrices are required to be SPD. However, even if the matrices are only positive semi definite, they can be regularized by adding a multiple of trace to diagonal entries of the covariance matrix:

$$\mathbf{C}^+ = \mathbf{C} + \lambda trace(\mathbf{C})\mathbf{I}, \tag{2}$$

where λ is a regularization parameter and $\mathbf{I}$ is identity matrix.

Covariance Matrix for Spatial Pooling: In order to apply covariance pooling to image-based facial expression recognition problem, as shown in Figure 2, outputs from final convolutional layers can be flattened and used to compute covariance matrix. Let $\mathbf{X} \in \mathbb{R}^{w \times h \times d}$ be the output obtained after several convolutional layers, where w, h, d stand for width, height and number of channels in the output respectively. $\mathbf{X}$ can be flattened as an element $\mathbf{X}' \in \mathbb{R}^{n \times d}$ where $n = w \times h$. If $\mathbf{f_1}, \mathbf{f_2}, ..., \mathbf{f_n} \in \mathbb{R}^d$ be columns of $\mathbf{X}'$, we can capture the variation across channels by computing covariance as in Eqn 1 and regularizing thus computed matrix using Eqn. 2.

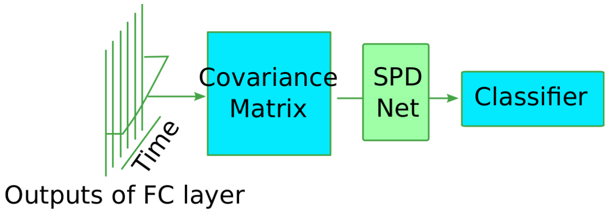


Figure 3. In case of video-based facial expression recognition problems, output of fully connected layers are considered as image set features. The covariance matrix is computed from such image set features.

Covariance Matrix for Temporal Pooling: As illustrated in Figure 3, covariance pooling can be employed in [17] to pool temporal features. If $\mathbf{f_1}, \mathbf{f_2}, \ldots, \mathbf{f_n} \in \mathbb{R}^d$ be per-frame features extracted from images, we can compute covariance matrix using the Eqn. 1 and regularize it using Eqn. 2.

3.3. SPD Manifold Network (SPDNet) Layers

The covariance matrices thus obtained typically reside on the Riemannian manifold of SPD matrices. Directly flattening and applying fully connected layers directly causes loss of geometric information. Standard methods apply logarithm operation to flatten the Riemannian manifold structure to be able to apply standard loss functions of Euclidean space [6][20]. The covariance matrices thus obtained are often large and their dimension needs to be reduced without losing geometric structure. In [11], authors introduce special layers for reducing dimension of SPD matrices and to flatten the Riemannian manifold to be able to apply standard loss functions.

In this subsection, we briefly discuss the layers introduced in [11] for learning on Riemannian Manifold.

Bilinear Mapping Layer (BiMap) Covariance matrices computed from features can be large and it may not be feasible to directly apply fully connected layers after flattening them. Furthermore, it is also important to preserve geometric structure while reducing dimension. The BiMap layer accomplishes both of these conditions and plays the same role as traditional fully connected layers. If $\mathbf{X}_{k-1}$ be input SPD matrix, $\mathbf{W}_k \in \mathbb{R}_*^{d_k \times d_{k-1}}$ be weight matrix in the space of full rank matrices and $\mathbf{X}_k \in \mathbb{R}^{d_k \times d_k}$ be output matrix, then k-th the bilinear mapping f_b^k is defined as

$$\mathbf{X}_k = f_b^k(\mathbf{X}_{k-1}; \mathbf{W}_k) = \mathbf{W}_k \mathbf{X}_{k-1} \mathbf{W}_k^T. \tag{3}$$

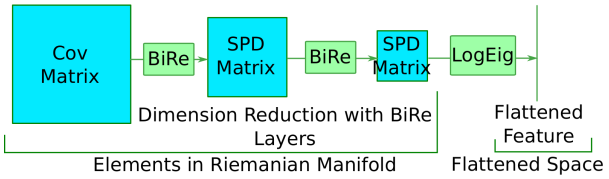


Figure 4. Illustration of SPD Manifold Network (SPDNet) with 2-BiRe layers.

Eigenvalue Rectification (ReEig) ReEig layer can be used to introduce non-linearity in the similar way as Rectified Linear Unit (ReLU) layers in traditional neural networks. If $\mathbf{X}_{k-1}$ be input SPD matrix, $\mathbf{X}_k$ be output and ϵ be eigenvalue rectification threshold, k-th ReEig Layer f_r^k is defined as:

$$\mathbf{X}_k = f_r^k(\mathbf{X}_{k-1}) = \mathbf{U}_{k-1}\max(\epsilon\mathbf{I}, \sigma_{k-1})\mathbf{U}_{k-1}^T, \quad (4)$$

where $\mathbf{U}_{k-1}$ and $\mathbf{\Sigma}_{k-1}$ are defined by eigenvalue decomposition $\mathbf{X}_{k-1} = \mathbf{U}_{k-1}\Sigma_{k-1}\mathbf{U}_{k-1}^T$. The $\max$ operation is element-wise matrix operation.

Log Eigenvalue Layer (LogEig) As discussed earlier, SPD matrices lie on Riemannian manifold. The final LogEig layer endows elements in Riemannian manifold with a Lie Group structure so that matrices can be flattened and standard euclidean operations can be applied. If $\mathbf{X}_{k-1}$ be input matrix, $\mathbf{X}_k$ be output matrix, the LogEig layer applied in k-th layer f_l^k is defined as

$$\mathbf{X}_k = f_l^k(\mathbf{X}_{k-1}) = \log(\mathbf{X}_{k-1}) = \mathbf{U}_{k-1}\log(\Sigma_{k-1})\mathbf{U}_{k-1}^T, \quad (5)$$

where $\mathbf{X}_k = \mathbf{U}_{k-1}\Sigma_{k-1}\mathbf{U}_{k-1}^T$ is an eigenvalue decomposition and $\log$ is an element-wise matrix operation.

BiMap and ReEig layers can be used together as a block and is abbreviated as BiRe. The architecture of a SPDNet with 2-BiRe layers is shown in Figure 4.

4. Experiments

4.1. Benchmark Datasets

Datasets that contain samples with either real or acted facial expressions in the wild were chosen. Such datasets are better approximation to the real world scenarios than posed datasets and are also more challenging.

Image-based Facial Expression Recognition For comparing our deep learning architectures for image-based facial expression recognition against standard results, we use Static Facial Expressions in the Wild (SFEW) 2.0 [2] [1] dataset and Real-world Affective Faces (RAF) dataset [16]. SFEW 2.0 contains 1394 images, of which 958 are to be used for training and 436 for validation. This dataset was prepared by selecting frames from videos of AFEW dataset. Facial landmark points provided by the authors were detected using mixture-of-parts based model [28]. The landmarks thus obtained were then used for alignment. The RAF dataset [16] was prepared by collecting images from various search engines and the facial landmarks were annotated manually by 40 independent labelers. The dataset contains 15331 images labeled with seven basic emotion categories of which 3068 are to be used for validation and 12271 for training.

It is worth pointing out that there exist several other image-based datasets such as EmotioNet [5] and FER-2013 [10]. However, they have their own downsides. Though EmotioNet is the largest existing dataset for facial expression recognition, the images were automatically annotated and the labels are incomplete. FER-2013, contains relatively small image size and does not contain RGB information. Most other databases either contain too few samples or are taken in well posed laboratory setting.

Video-based Facial Expression Recognition For video-based facial expression recognition, we use Acted Facial Expressions in the Wild (AFEW) dataset to compare our methods with existing methods. This dataset was prepared by selecting videos from movies. It contains about 1156 publicly available labeled videos of which 773 videos are used for training and 383 for validation. Just as in case of SFEW 2.0 dataset, the landmarks and aligned images provided by authors were obtained using mixture-of-parts based model.

Though there exist several other facial expression recognition databases for videos such as Cohn-Kanade/Cohn-Kanade+ (CK/CK+) [14][18], most of them are either sampled in well controlled laboratory environment or are labeled with action unit encoding rather than seven basic classes of facial expressions.

4.2. Face Detection and Alignment

Authors of RAF database [16] provide manually annotated face landmarks, while those of SFEW 2.0 [2] and AFEW [1] datasets do not and instead provide landmarks and aligned images obtained using mixture-of-parts based model [28]. Images and videos captured in the wild contain large amount of non-essential information. Face detection and alignment helps remove non-essential information from the data samples. Furthermore, to be able to compare variations in local facial features across images, face alignment is important. This serves as normalization of data. While trying to categorize facial expressions from videos, motion of people, change of background etc. also lead to large unwanted variation across image frames. Due to this, training algorithms on original unaligned data is not feasible. Face alignment additionally helps to capture the dynamic evolu-

Models	RAF Total	SFEW 2.0 Total
VGG-VD-16 network[3]	-	54.82
Inception-ResnetV1 (Trained from scratch)‡	82.6	47.37
Inception-ResnetV1 (Finetuned) ‡	83.4	51.9
Baseline Model ‡	84.5	54.45

Table 1. Comparison of image-base recognition accuracies of various standard models on validation set of the RAF and SFEW 2.0 datasets. Here the models labelled ‡ were trained on our own.

tion of local facial features across images of the same videos in an effective manner.

For face and facial landmark detection Multi-task Cascade Convolutional Neural Networks (MTCNN) [27] was used. MTCNN was found to be more accurate and successful for alignment compared to mixture-of-parts based model. After successful face and facial landmark detection, we use three points constrained affine transformation for face alignment. Coordinates of left eye, right eye and midpoint of corners of the lips were used for alignment.

4.3. Baseline Model and Architectures for Image-based Problem

Comparison of Standard Architectures In Table 1 we present the comparison of accuracies of training or finetuning various standard network architectures. For a baseline model, we take the network architecture presented in [16]. The scores reported on RAF database for VGG network and AlexNet in [16] is less compared to their base line model. So the networks are not trained again here. It is worth pointing out that there, authors report per class average accuracy but we report total accuracy only here. Here, we use center loss[22] to train the network in all cases rather than locality preserving loss[16] as we do not deal with compound emotions. In all cases, dataset was augmented using standard techniques such as random crop, random rotate and random flip. For SFEW 2.0, in all cases, output of second to last fully connected layer was used as image features and Support Vector Machines (SVMs) were trained separately. Note that the models labelled ‡ were trained on our own. Inception-ResnetV1 [19] was both trained from scratch, as well as finetuned on a model pre-trained on subset of MS-Celeb-1M dataset. It is evident from the table that fine-tuning the Inception-ResnetV1 trained on face recognition dataset performs better than training from scratch. It should not come as a surprise that a relatively small network outperforms Inception-ResNet model as there are more parameters to be learned in deeper models. For further experiments and to introduce covariance pooling, we use the baseline model from [16].

Incorporation of SPD Manifold Network As discussed above, we introduce covariance pooling and subsequently the layers from the SPD manifold network (SPDNet) after the final convolutional layer. While introducing covariance pooling, we experimented with various models for the architecture. The details of the various models considered are summarized in Table 2.

Baseline	Mode- 1	Model-2	Model-3	Model-4
Conv256	Conv256	Conv256	Conv256	Conv256
	Cov BiRe LogEig	Cov BiRe LogEig	Cov BiRe BiRe LogEig	Cov BiRe LogEig
FC2000 FC7	FC2000 FC7	FC2000 FC128 FC7	FC2000 FC7	FC2000 FC512 FC7

Table 2. Various models considered for covariance pooling. For brevity, initial convolution layers are ignored.

4.4. Results on Image-based Problem

Covariance pooling was applied after final convolution layer and before fully connected layers. Various models described in Table 2 and their accuracies are listed below in Table 3. For the RAF database, as stated earlier, the network was trained in end-to-end fashion. However, for SFEW 2.0 dataset, we use output of penultimate fully connected layer (which ranges from 128 to 2000 dimensional feature depending on the model considered). It is worth pointing out that for SFEW 2.0 our single model performed better than ensemble of convolutional neural networks in [26] and [15]. It could be argued that the datasets used for pre-training were different in our case and in [26][15]. However, improvement of almost 3.7% over baseline in the

Model	RAF Total Accuracy	SFEW 2.0 Total Accuracy
Baseline Model [16]	84.7	54.45
Model-1	86.3	55.40
Model-2	**87.0**	56.72
Model-3	85.0	57.48
Model-4	85.4	**58.14**
VGG-VD-16 [3]	-	54.82
EmotiW-1 (2015) [26]	-	55.96
EmotiW-2 (2015) [15]	-	52.80

Table 3. Image-based recognition accuracies for various models with and without covariance pooling.

SFEW 2.0 dataset justifies the use of SPDNet for facial expression recognition.

It is also important to point out that on the SFEW 2.0 and AFEW datasets, face detection failed in several images and videos. To report validation score, we assign random uniform probability of success ($\frac{1}{7}$) for correct recognition to the samples on which face detection did not succeed.

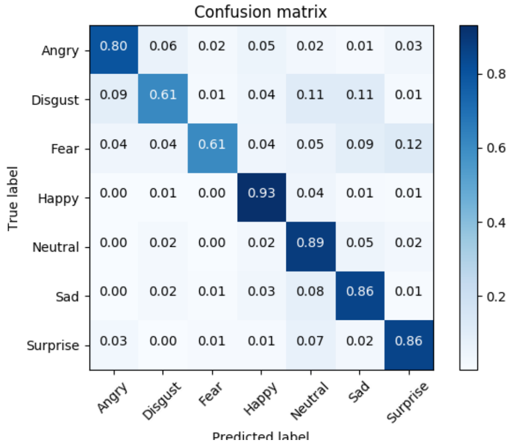


Figure 5. Confusion matrix for Model-2 on the RAF dataset.

4.5. Baseline Model for Video-based Recognition Problem

For comparing the benefits of using SPDNet over existing methods, we use kernel based PLS that used covariance matrices as features [17] in baseline method. 128 dimensional features were extracted from each image frame of a video and the video was modeled with a covariance matrix. Then either SPDNet or kernel based SVM with either RBF or Polynomial kernel were used for recognition. The SPDNet was able to outperform other methods.

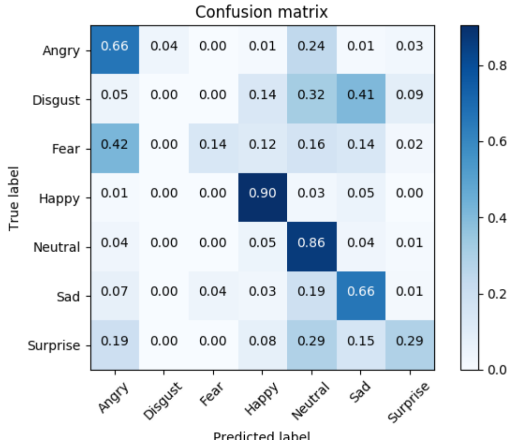


Figure 6. Confusion matrix for Model-4 on the SFEW 2.0 dataset.

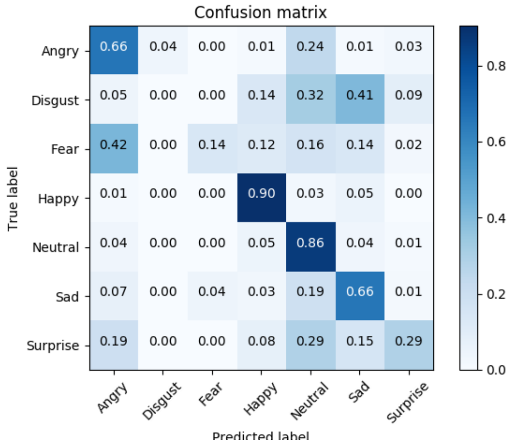


Figure 7. Confusion matrices for our method (4-Bire) on the AFEW dataset.

4.6. Results on Video-based Problem

The results of our proposed methods, baseline method and the accuracies of other C3D and CNN-RNN models from [9] are presented for context. However, datasets used for those pretraining other models are not uniform, and detailed comparison of all existing methods is not within the scope of this work. As seen from Table 5, our model was able to slightly surpass the results of the base line model. Our method also performed better than all single models that were trained on publicly available training dataset. The network from [4] was trained on private dataset containing an order of magnitude more samples. As a side experimentation, we also introduced covariance pooling to C3D model in [9] and did not obtain any improvement. We obtained accuracy of 39.78%.

5. Conclusion

In this work, we exploit the use of SPDNet on facial expression recognition problems. As shown above, SPDNet applied to covariance of convolutional features can classify facial expressions more efficiently. We study that second-order networks are better able to capture facial landmark distortions. Similarly, covariance matrix computed from image feature vectors were used as input to SPDNet for video-based facial expression recognition problem.

We were able to obtain state-of-the-art results on image-based facial expression recognition problems on the SFEW 2.0 and RAF datasets. In video-based facial expression recognition, training SPDNet on image-based features was able to obtain results comparable to state-of-the-art results.

In the context of video-based facial expression recognition problem, architecture presented in Figure 8 can be trained in end-to-end training. Though with brief experimentation, we were able to obtain accuracy of only 32.5%

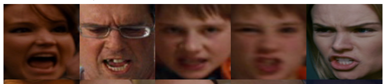
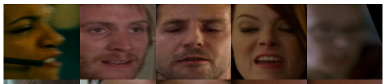
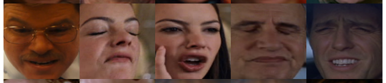
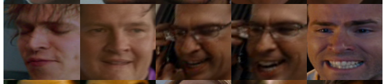
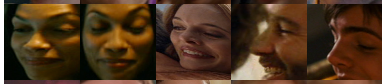
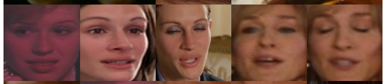
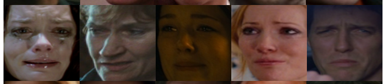
Original Class	Correctly Predicted	Incorrectly Predicted	Predicted Classes
Angry			Neutral, Neutral, Neutral, Neutral, Happy
Disgust			Sad, Sad, Surprise, Sad, Neutral
Fear			Happy, Happy, Neutral, Angry, Happy
Happy			Sad, Neutral, Neutral, Sad, Angry
Neutral			Angry, Happy, Happy, Happy, Happy
Sad			Neutral, Angry, Happy, Surprise, Neutral
Surprise			Happy, Happy, Happy, Neutral, Happy

Table 4. Samples from each class of the SFEW dataset that were most accurately and least accurately classified. The first column indicates ground truth and final column indicates predicted labels for incorrectly predicted images.

Model	AFEW
VGG13 [4]	57.07
Single Best CNN-RNN [9]	45.43
Single Best C3D [9]	39.69
Single Best HoloNet [23]	44.57
Baseline (RBF Kernel) [17]	45.95
Baseline (Poly Kernel) [17]	45.43
Our proposed method (2-Bire)	42.25
Our proposed method (3-Bire)	44.09
Our proposed method (4-Bire)	46.71
Multiple CNN-RNN and C3D ** [9]	51.8
VGG13+VGG16+ResNet ** [23]	59.16

Table 5. Video-based recognition accuracies for various single models and fusion of multiple models. Here the results of the methods marked with ** were obtained either by score level or feature level fusion of multiple models.

which is worse than the score reported [11]. It is likely to be a result of relatively small size of AFEW dataset compared to parameters in the network. Further work is necessary to see if training end-to-end using joint convolutional net and SPD net can improve results.

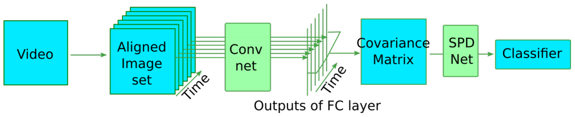


Figure 8. Architecture for end-to-end training on videos directly.

6. Further Works

In this work, we leveraged covariance matrix to capture second-order statistics. As studied in [12], Gaussian matrix is able to further improve the effectiveness of second-order statistics. Formally, the SPD form of Gaussian matrix can be computed by

$$\mathbf{G} = \begin{pmatrix} \mathbf{\Sigma} + \mu\mu^{\mathbf{T}} & \mu \\ \mu^{\mathbf{T}} & 1 \end{pmatrix}, \quad (6)$$

where $\mathbf{\Sigma}$ is the covariance matrix defined in Eqn. 1, and

$$\mu = \sum_{i=1}^{n} \mathbf{f_i} \quad (7)$$

is the mean of the samples $\mathbf{f_1}, \mathbf{f_2}, \ldots, \mathbf{f_n}$ captures both first-order and second-order statistics. It was also employed in [25] to develop second-order convolutional neural networks. Extending current work from covariance pooling to Gaussian pooling would be an interesting direction and should theoretically improve results.

References

[1] J. J. K. S. A. Dhall, R. Goecke and T. Gedeon. Emotion recognition in the wild challenge 2014: Baseline, data and protocol. In *ACM ICMI*, 2014. 4

[2] S. L. A. Dhall, R. Goecke and T. Gedeon. Collecting large, richly annotated facialexpression databases from movies. In *IEEE MultiMedia 19*, 2012. 4

[3] S. Albanie and A. Vedaldi. Learning grimaces by watching tv. In *Proceedings of the British Machine Vision Conference (BMVC)*, 2016. 2, 5

[4] S. A. Bargal, E. Barsoum, C. C. Ferrer, and C. Zhang. Emotion recognition in the wild from videos using images. In *Proceedings of the 18th ACM International Conference on Multimodal Interaction*, ICMI 2016, pages 433–436, New York, NY, USA, 2016. ACM. 2, 6, 7

[5] C. F. Benitez-Quiroz, R. Srinivasan, and A. M. Martinez. Emotionet: An accurate, real-time algorithm for the automatic annotation of a million facial expressions in the wild. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5562–5570, June 2016. 4

[6] J. a. Carreira, R. Caseiro, J. Batista, and C. Sminchisescu. Semantic segmentation with second-order pooling. In *Proceedings of the 12th European Conference on Computer Vision - Volume Part VII*, ECCV'12, pages 430–443, Berlin, Heidelberg, 2012. Springer-Verlag. 2, 3

[7] H. Ding, S. K. Zhou, and R. Chellappa. Facenet2expnet: Regularizing a deep face recognition net for expression recognition. In *2017 12th IEEE International Conference on Automatic Face Gesture Recognition (FG 2017)*, pages 118–126, May 2017. 2

[8] S. Ebrahimi Kahou, V. Michalski, K. Konda, R. Memisevic, and C. Pal. Recurrent neural networks for emotion recognition in video. In *Proceedings of the 2015 ACM on International Conference on Multimodal Interaction*, ICMI '15, pages 467–474, New York, NY, USA, 2015. ACM. 2

[9] Y. Fan, X. Lu, D. Li, and Y. Liu. Video-based emotion recognition using cnn-rnn and c3d hybrid networks. In *Proceedings of the 18th ACM International Conference on Multimodal Interaction*, ICMI 2016, pages 445–450, New York, NY, USA, 2016. ACM. 2, 6, 7

[10] I. J. Goodfellow, D. Erhan, P. Luc Carrier, A. Courville, M. Mirza, B. Hamner, W. Cukierski, Y. Tang, D. Thaler, D.-H. Lee, Y. Zhou, C. Ramaiah, F. Feng, R. Li, X. Wang, D. Athanasakis, J. Shawe-Taylor, M. Milakov, J. Park, R. Ionescu, M. Popescu, C. Grozea, J. Bergstra, J. Xie, L. Romaszko, B. Xu, Z. Chuang, and Y. Bengio. Challenges in representation learning. *Neural Netw.*, 64(C):59–63, Apr. 2015. 4

[11] Z. Huang and L. V. Gool. A riemannian network for spd matrix learning. In *AAAI*, 2017. 1, 2, 3, 7

[12] Z. Huang, R. Wang, S. Shan, X. Li, and X. Chen. Log-euclidean metric learning on symmetric positive definite manifold with application to image set classification. In *ICML*, pages 720–729, 2015. 7

[13] C. Ionescu, O. Vantzos, and C. Sminchisescu. Matrix backpropagation for deep networks with structured layers. *2015 IEEE International Conference on Computer Vision (ICCV)*, pages 2965–2973, 2015. 1

[14] T. Kanade, J. F. Cohn, and Y. Tian. Comprehensive database for facial expression analysis. In *Proceedings Fourth IEEE International Conference on Automatic Face and Gesture Recognition (Cat. No. PR00580)*, pages 46–53, 2000. 4

[15] B.-K. Kim, H. Lee, J. Roh, and S.-Y. Lee. Hierarchical committee of deep cnns with exponentially-weighted decision fusion for static facial expression recognition. In *Proceedings of the 2015 ACM on International Conference on Multimodal Interaction*, ICMI '15, pages 427–434, New York, NY, USA, 2015. ACM. 2, 5

[16] S. Li, W. Deng, and J. Du. Reliable crowdsourcing and deep locality-preserving learning for expression recognition in the wild. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, July 2017. 4, 5

[17] M. Liu, R. Wang, S. Li, S. Shan, Z. Huang, and X. Chen. Combining multiple kernel methods on riemannian manifold for emotion recognition in the wild. In *Proceedings of the 16th International Conference on Multimodal Interaction*, ICMI '14, pages 494–501, New York, NY, USA, 2014. ACM. 1, 2, 3, 6, 7

[18] P. Lucey, J. F. Cohn, T. Kanade, J. Saragih, Z. Ambadar, and I. Matthews. The extended cohn-kanade dataset (ck+): A complete dataset for action unit and emotion-specified expression. In *2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition - Workshops*, pages 94–101, June 2010. 4

[19] C. Szegedy, S. Ioffe, V. Vanhoucke, and A. A. Alemi. Inception-v4, inception-resnet and the impact of residual connections on learning. In *AAAI*, 2017. 5

[20] O. Tuzel, F. Porikli, and P. Meer. Region covariance: A fast descriptor for detection and classification. In *Proceedings of the 9th European Conference on Computer Vision - Volume Part II*, ECCV'06, pages 589–600, Berlin, Heidelberg, 2006. Springer-Verlag. 1, 2, 3

[21] F. Wang, X. Xiang, C. Liu, T. D. Tran, A. Reiter, G. D. Hager, H. Quon, J. Cheng, and A. L. Yuille. Regularizing face verification nets for pain intensity regression. *2017 IEEE International Conference on Image Processing (ICIP)*, pages 1087–1091, 2017. 2

[22] Y. Wen, K. Zhang, Z. Li, and Y. Qiao. *A Discriminative Feature Learning Approach for Deep Face Recognition*, pages 499–515. Springer International Publishing, Cham, 2016. 5

[23] J. Yan, W. Zheng, Z. Cui, C. Tang, T. Zhang, Y. Zong, and N. Sun. Multi-clue fusion for emotion recognition in the wild. In *Proceedings of the 18th ACM International Conference on Multimodal Interaction*, ICMI 2016, pages 458–463, New York, NY, USA, 2016. ACM. 2, 7

[24] A. Yao, J. Shao, N. Ma, and Y. Chen. Capturing au-aware facial features and their latent relations for emotion recognition in the wild. In *Proceedings of the 2015 ACM on International Conference on Multimodal Interaction*, ICMI '15, pages 451–458, New York, NY, USA, 2015. ACM. 2

[25] K. Yu and M. Salzmann. Second-order convolutional neural networks. *CoRR*, abs/1703.06817, 2017. 1, 2, 3, 7

[26] Z. Yu and C. Zhang. Image based static facial expression recognition with multiple deep network learning. In *Proceedings of the 2015 ACM on International Conference on Multimodal Interaction*, ICMI '15, pages 435–442, New York, NY, USA, 2015. ACM. 2, 5

[27] K. Zhang, Z. Zhang, Z. Li, and Y. Qiao. Joint face detection and alignment using multitask cascaded convolutional networks. *IEEE Signal Process. Lett.*, 23(10):1499–1503, 2016. 5

[28] X. Zhu and D. Ramanan. Face detection, pose estimation and landmark estimation in the wild. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2012. 4