UNIVERSITY OF WOLLONGONG AUSTRALIA

Learning SPD-matrix-based Representation for Visual Recognition

Lei Wang
VILA group
School of Computing and Information Technology
University of Wollongong, Australia

02-Nov-2019

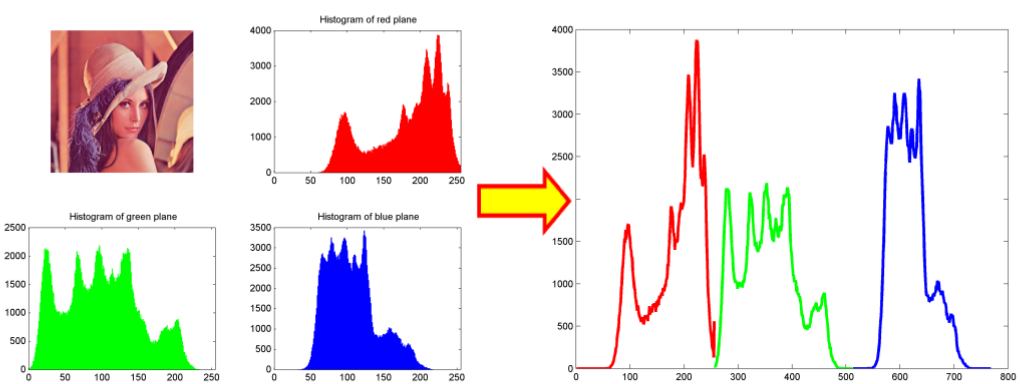
Introduction

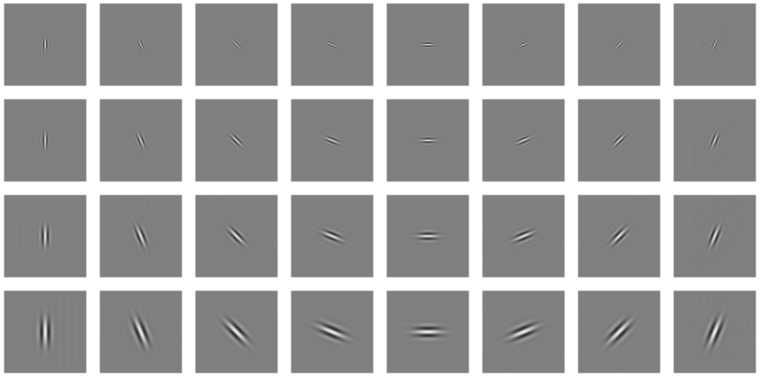
- How to **represent** an image?
  - Scale, rotation, illumination, occlusion, background clutter, deformation, …

Cat:

1. Before year 2000

- Hand-crafted, **global** features
  - Color, texture, shape, structure, etc.
  - Goal: "**Invariant and discriminative**"
- Classifier
  - K-nearest neighbor, SVMs, Boosting, …



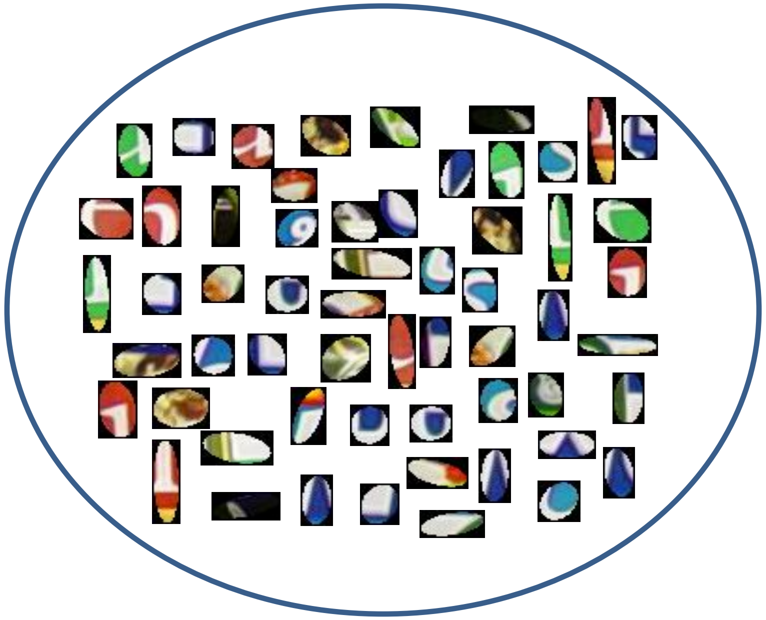


2. Days of the Bag of Features (BoF) model

Local Invariant Features

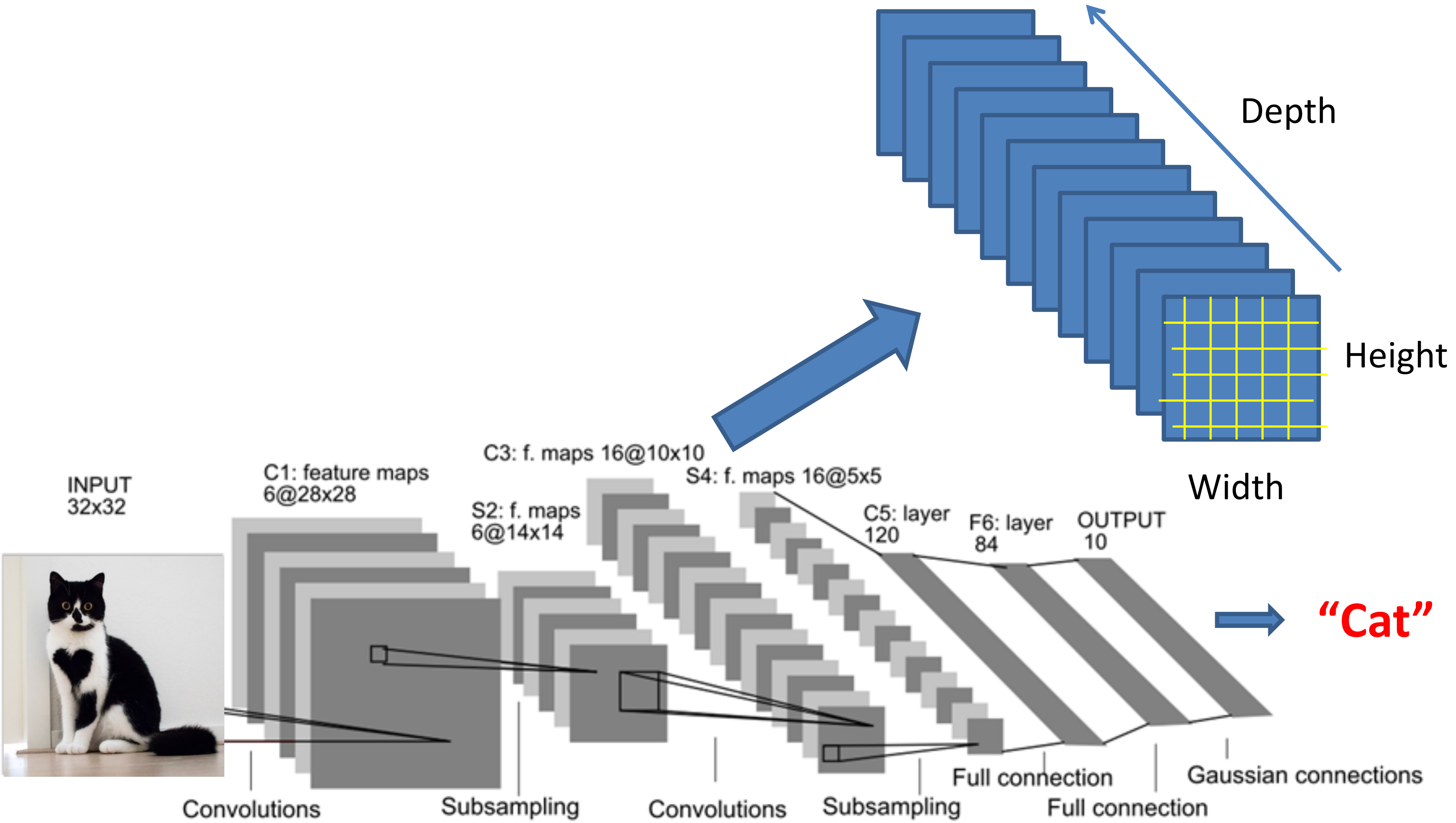
- **Invariant** to view angle, rotation, scale, illumination, clutter, ...

Interest point detection or Dense sampling



An image becomes "A bag of features"

Deep Local Descriptors



Image(s): a set of points/vectors

Object detection & classification

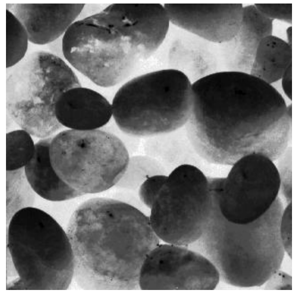
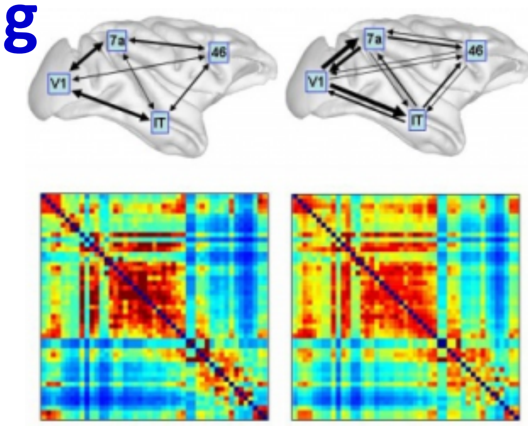


Image set classification

vs.

Action recognition

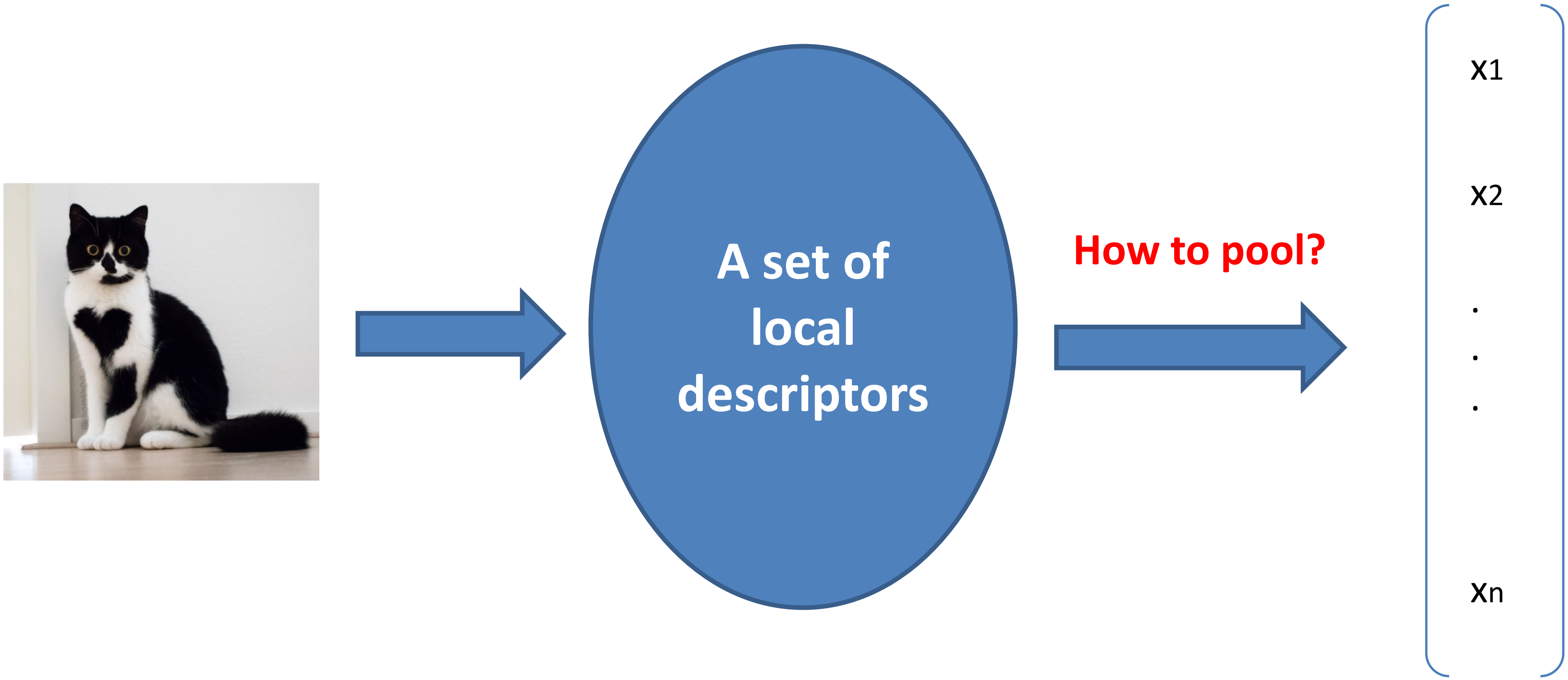
Neuroimaging analysis



How to pool a set of points/vectors to obtain a global visual representation ?

Covariance representation

Essentially a second-order pooling

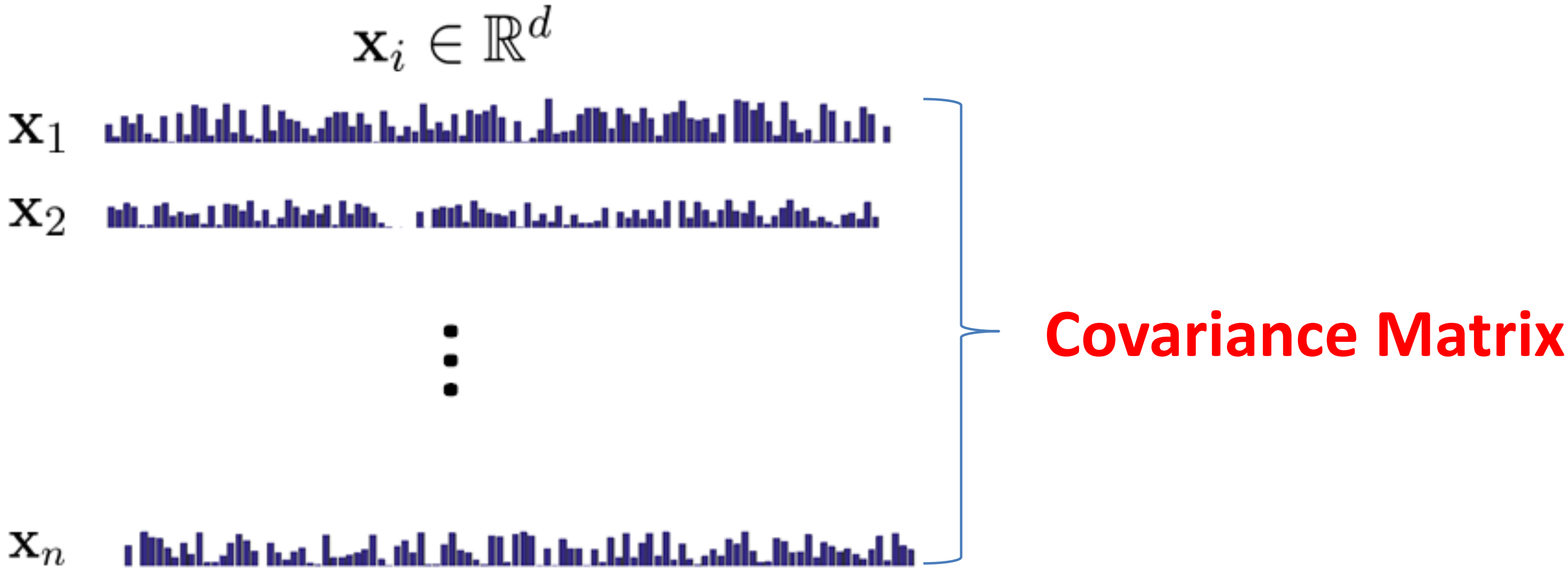


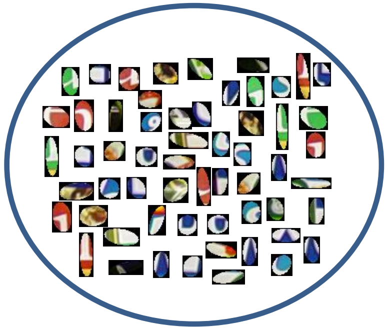
- Max pooling, average (sum) pooling, etc.
- Covariance pooling

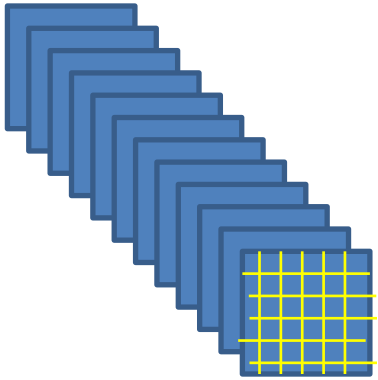
Outline

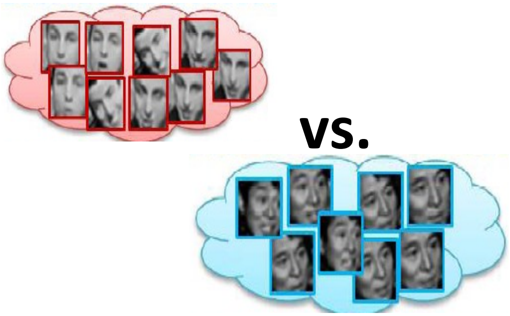
- Introduction on **Covariance** representation
- Our research work
  - **Discriminatively Learning** Covariance Representation
  - **Exploring Sparse** Inverse Covariance Representation
  - **Moving to Kernel-matrix**-based Representation (KSPD)
  - **Learning KSPD in deep** neural networks
- Conclusion

Introduction on Covariance representation





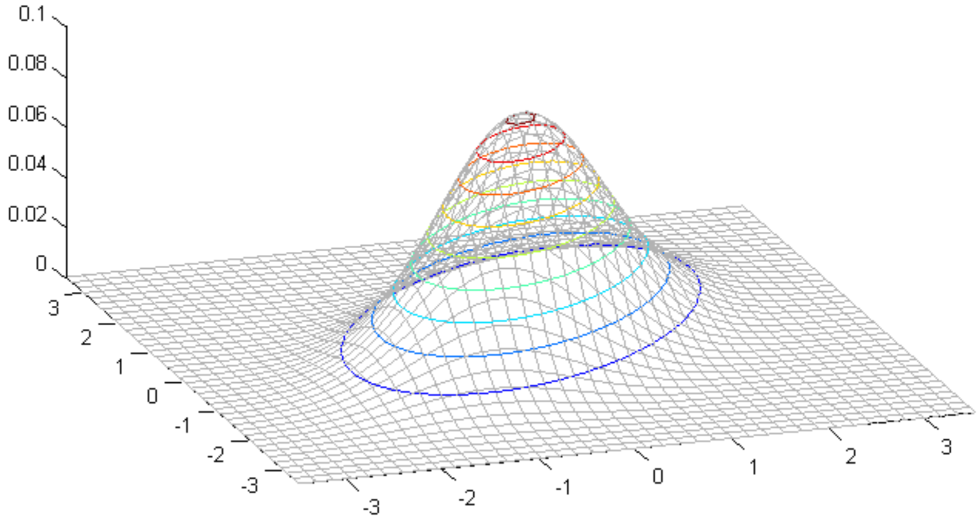




Introduction on Covariance representation

Use a **Covariance matrix** as a feature representation

$$\boldsymbol{x}_{d\times 1} \sim \mathcal{N}(\boldsymbol{\mu}_{d\times 1}, \boldsymbol{\Sigma}_{d\times d})$$



$$\boldsymbol{\mu} = \frac{1}{n}\sum_{i=1}^{n} \boldsymbol{x}_i$$

$$\boldsymbol{\Sigma} = \frac{1}{n-1}\sum_{i=1}^{n} (\boldsymbol{x}_i - \boldsymbol{\mu})(\boldsymbol{x}_i - \boldsymbol{\mu})^\top$$

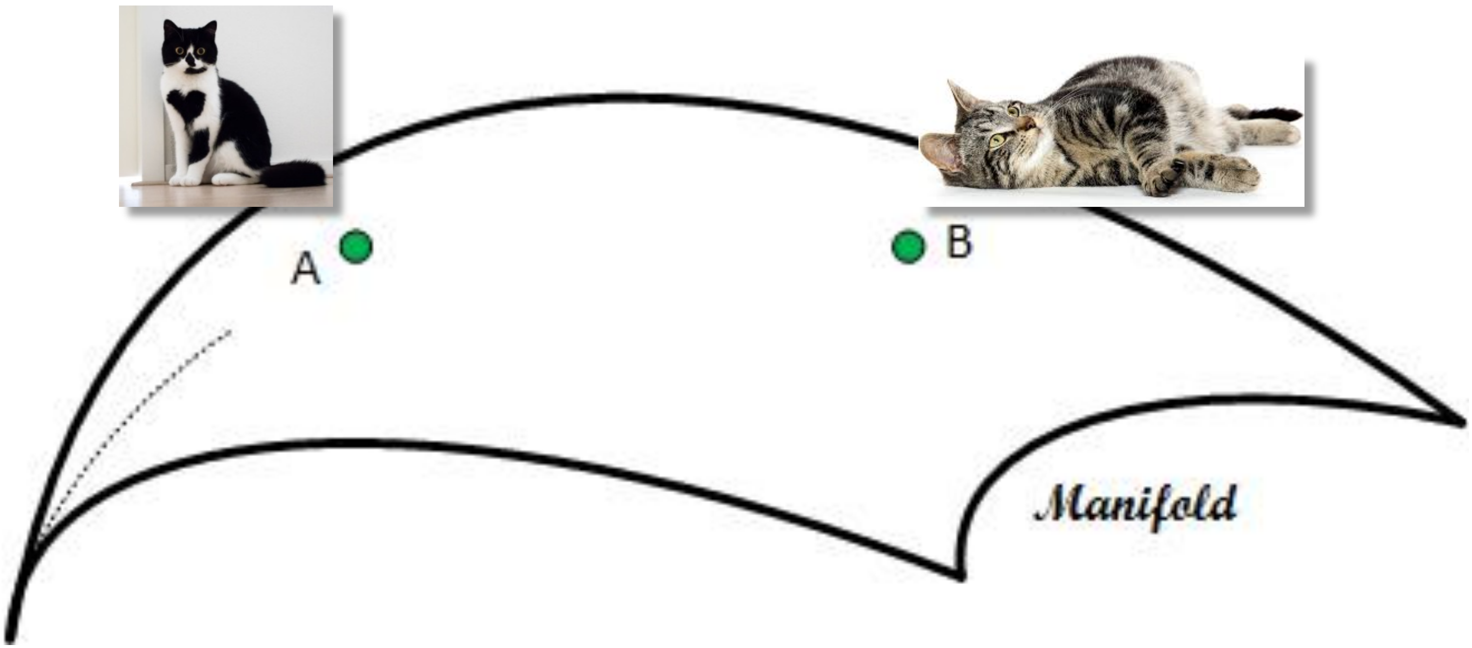
$$\boldsymbol{\Sigma} = \begin{pmatrix} \sigma_1^2 & \sigma_{12} & \sigma_{13} \\ \sigma_{21} & \sigma_2^2 & \sigma_{23} \\ \sigma_{31} & \sigma_{32} & \sigma_3^2 \end{pmatrix}$$

Image is from http://www.statsref.com/HTML/index.html?multivariate_distributions.html

Introduction on Covariance representation

$\boldsymbol{\Sigma}$ belongs to **Symmetric Positive Definite** (SPD) matrix

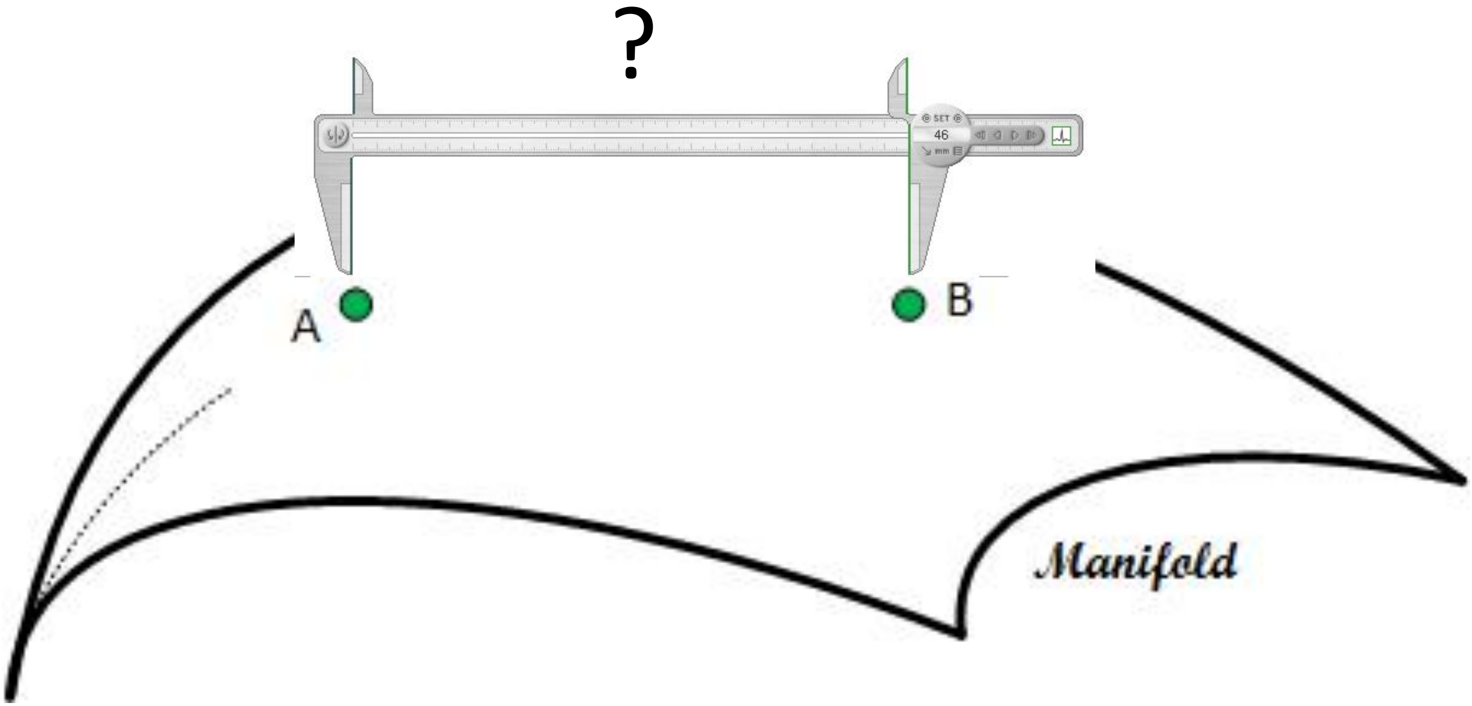
$$\mathrm{Sym}_d^+ = \{\boldsymbol{A} | \boldsymbol{A} = \boldsymbol{A}^\top, \forall \boldsymbol{x} \in \mathbb{R}^d, \boldsymbol{x} \neq \mathbf{0}, \boldsymbol{x}^\top \boldsymbol{A}\boldsymbol{x} > 0\}$$



$\boldsymbol{\Sigma}$ resides on a **manifold** instead of the whole space

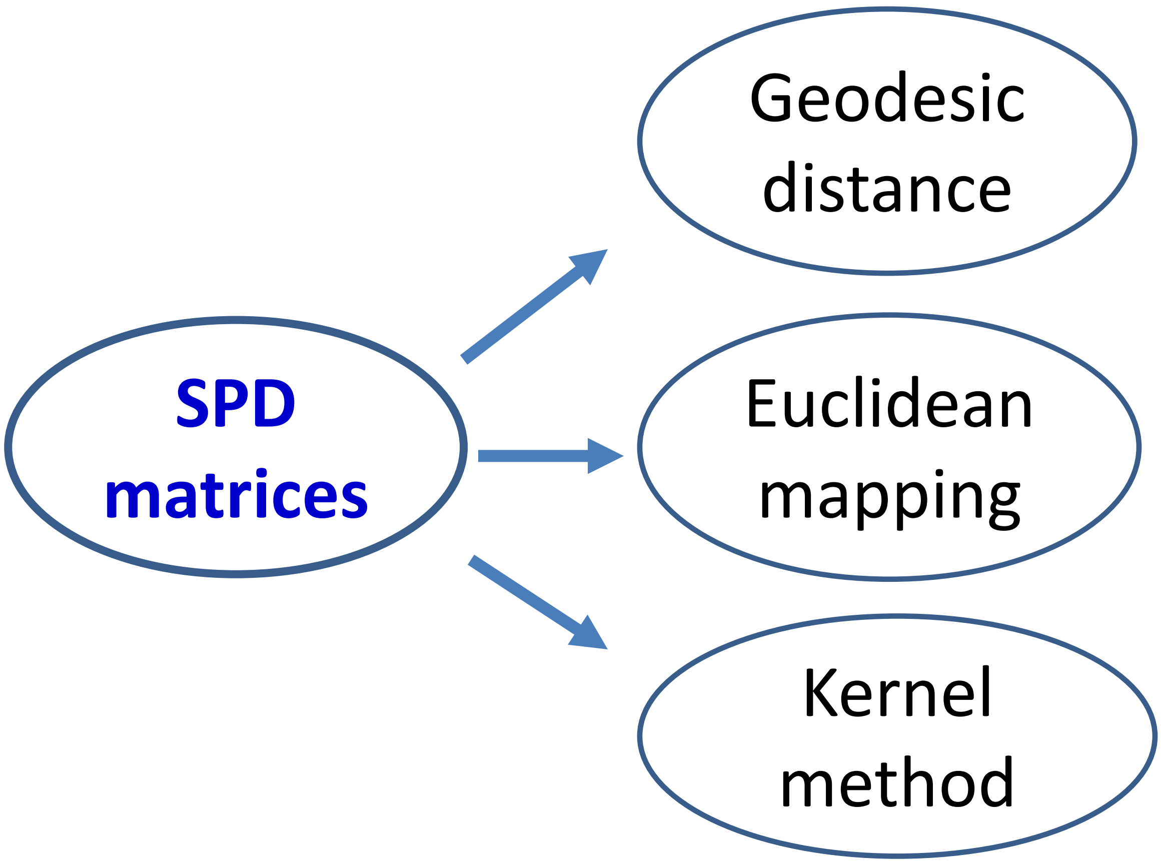
Introduction on Covariance representation

How to **measure the similarity** of two SPD matrices?

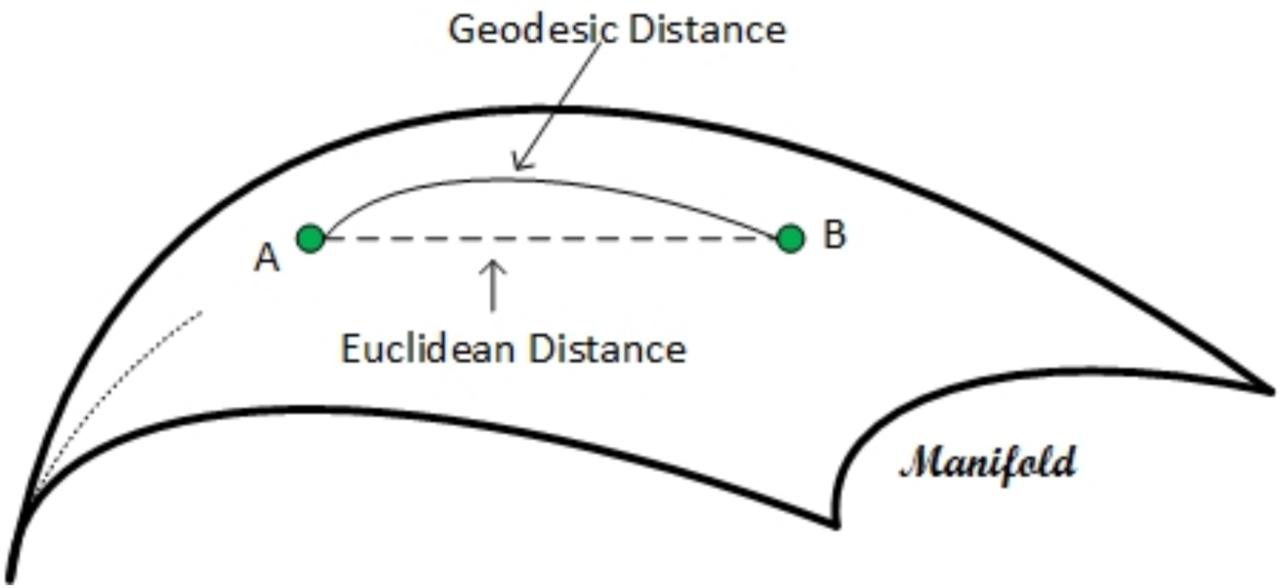


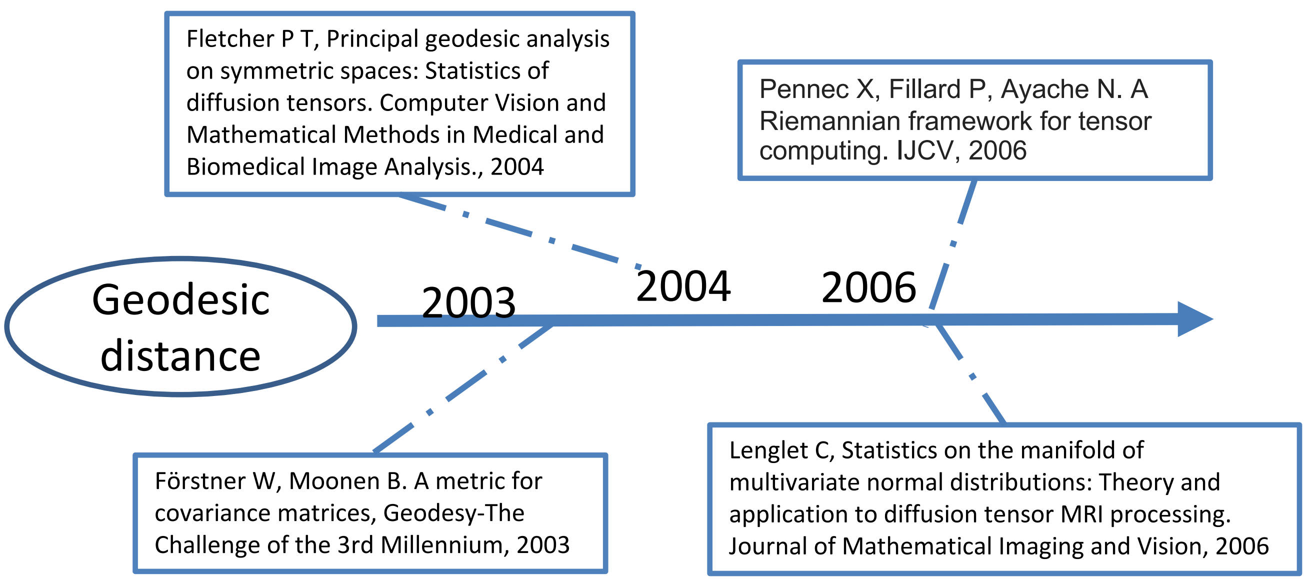
Introduction on SPD matrix

Similarity measures for SPD matrices

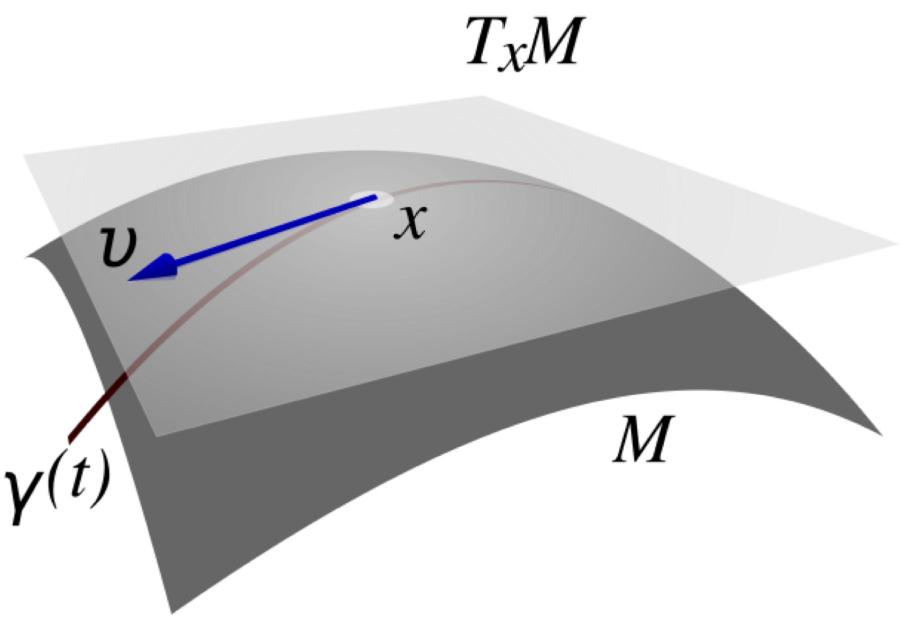


Introduction on SPD matrix





Introduction on SPD matrix



Euclidean mapping

2005

2006

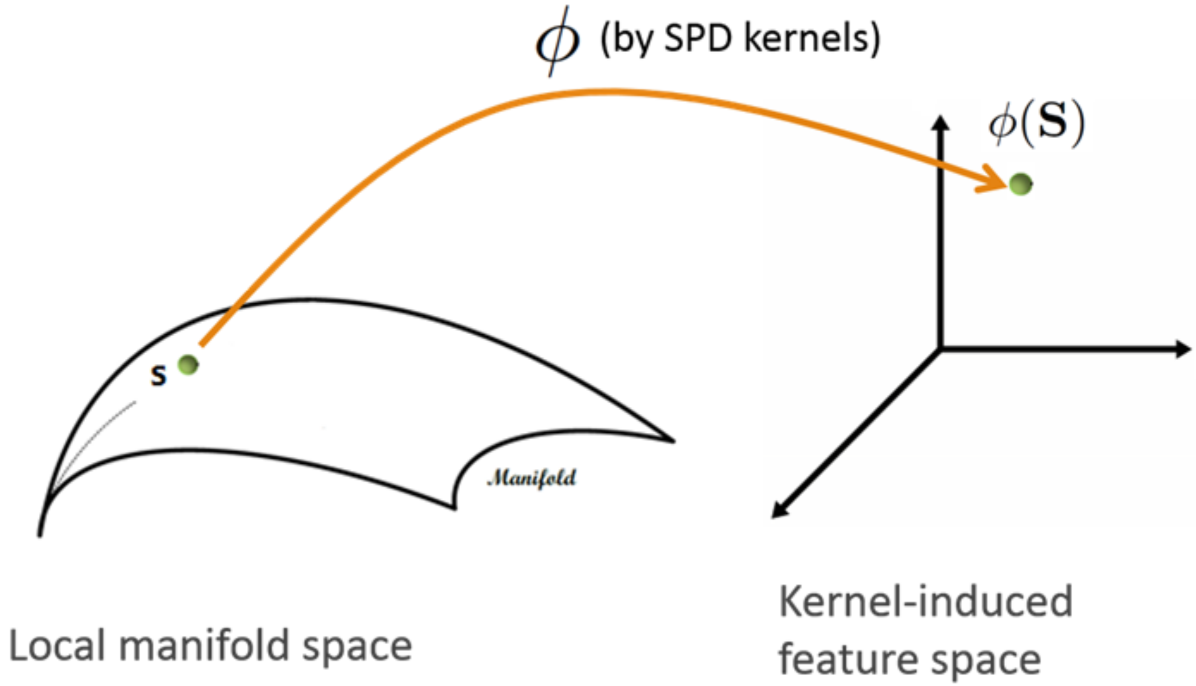
2008

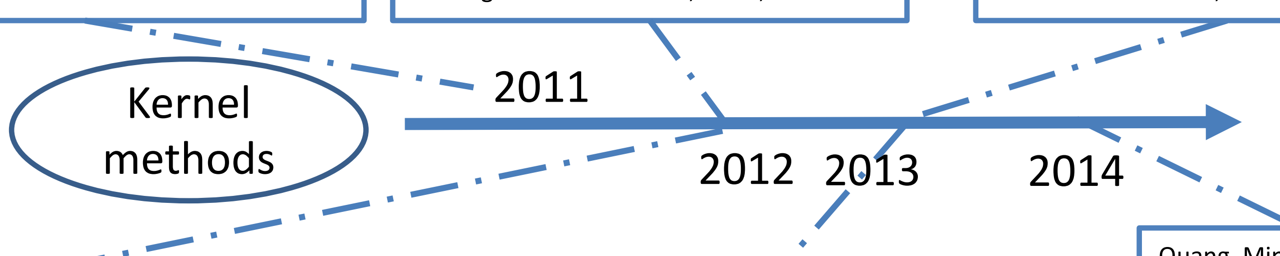
Arsigny V, Log-Euclidean metrics for fast and simple calculus on diffusion tensors. Magnetic resonance in medicine, 2006,

Veeraraghavan A, Matching shape sequences in video with applications in human movement analysis. PAMI, IEEE Transactions on, 2005

Tuzel O, Pedestrian detection via classification on riemannian manifolds. PAMI, IEEE Transactions on, 2008

Introduction on SPD matrix





Sra S. Positive definite matrices and the S-divergence. arXiv preprint arXiv:1110.1773, 2011.

Wang R., et. al., Covariance discriminative learning: A natural and efficient approach to image set classification, CVPR, 2012

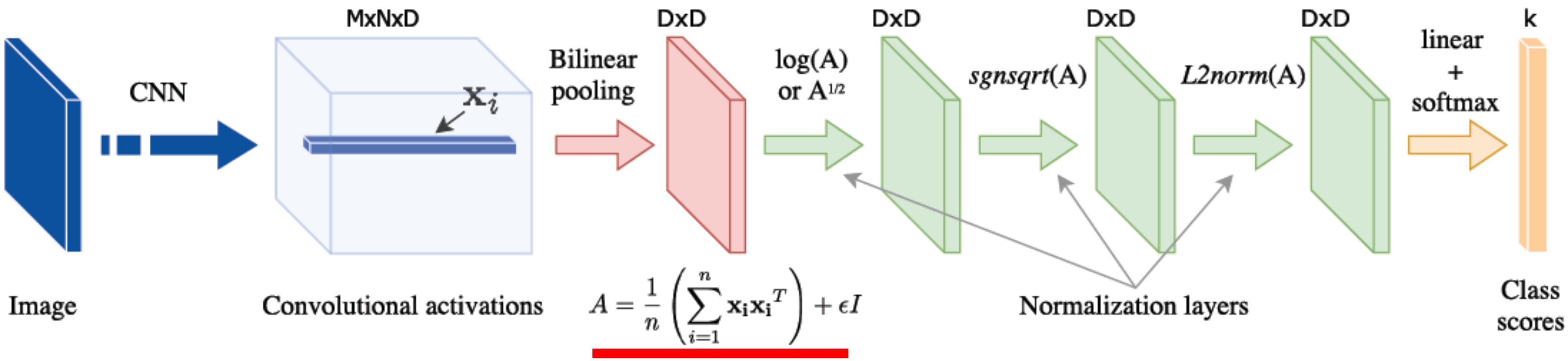
Vemulapalli R, Pillai J K, Chellappa R. Kernel learning for extrinsic classification of manifold features, CVPR, 2013

Harandi M et al. Sparse coding and dictionary learning for SPD matrices: a kernel approach, ECCV, 2012

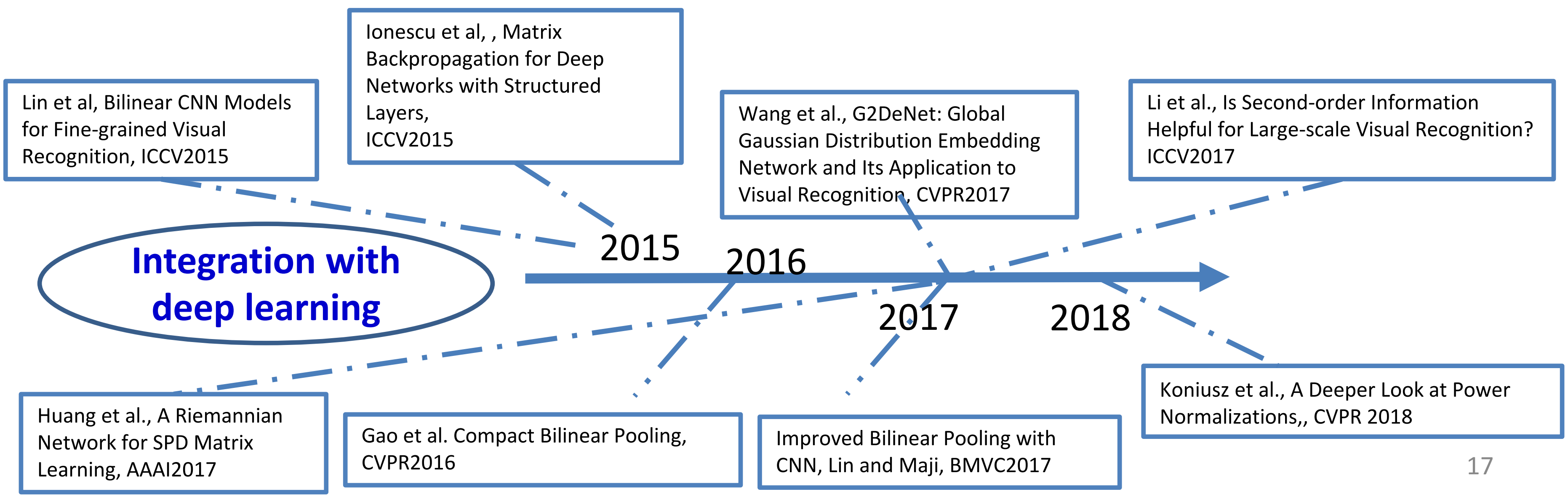
S. Jayasumana, et. al., Kernel methods on the Riemannian manifold of symmetric positive definite matrices, CVPR 2013.

Quang, Minh Ha, et. Al., Log-Hilbert-Schmidt metric between positive definite operators on Hilbert spaces. NIPS. 2014.

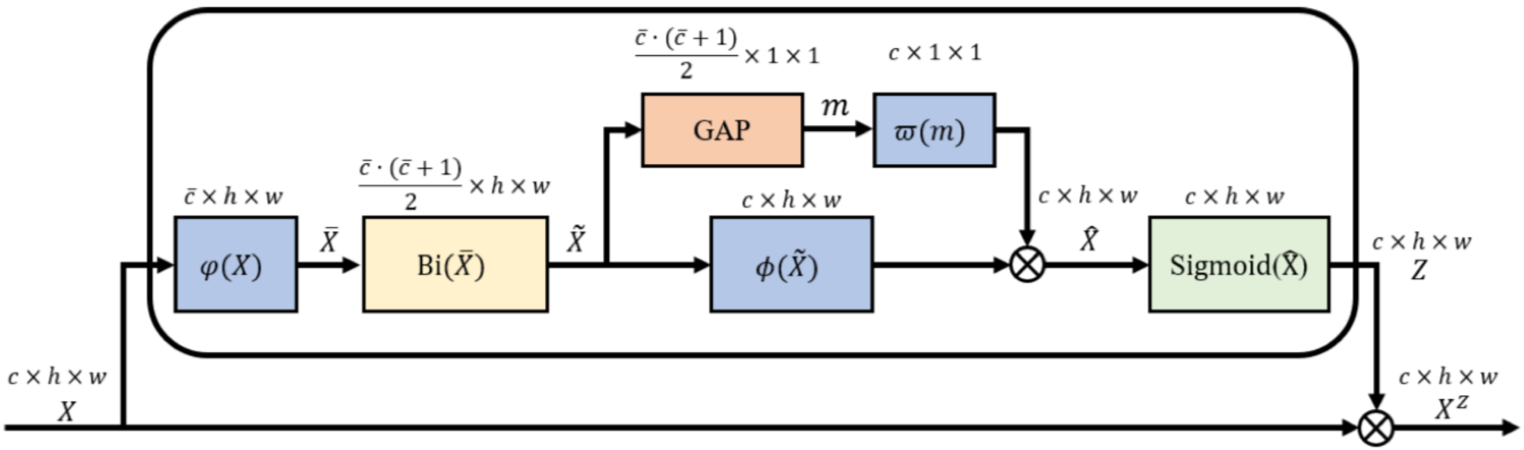
Introduction on SPD matrix



Improved Bilinear Pooling with CNN, Lin and Maji, BMVC2017



Introduction on SPD matrix



Fang et al., Bilinear Attention Networks for Person Retrieval, ICCV2019

Integration with deep learning

2018

Li et al, Towards Faster Training of Global Covariance Pooling Networks by Iterative Matrix Square Root Normalization, CVPR2018

Wei et al., Kernelized Subspace Pooling for Deep Local Descriptors, CVPR2018

Yu and Salzmann, Statistically-motivated Second-order Pooling, ECCV2018

Yu et al., Hierarchical Bilinear Pooling for Fine-Grained Visual Recognition, ECCV2018

Lin et al., Second-order Democratic Aggregation, ECCV2018

2019

Gao et al., Global Second-order Pooling Convolutional Networks, CVPR2019

Fang et al., Bilinear Attention Networks for Person Retrieval, ICCV2019

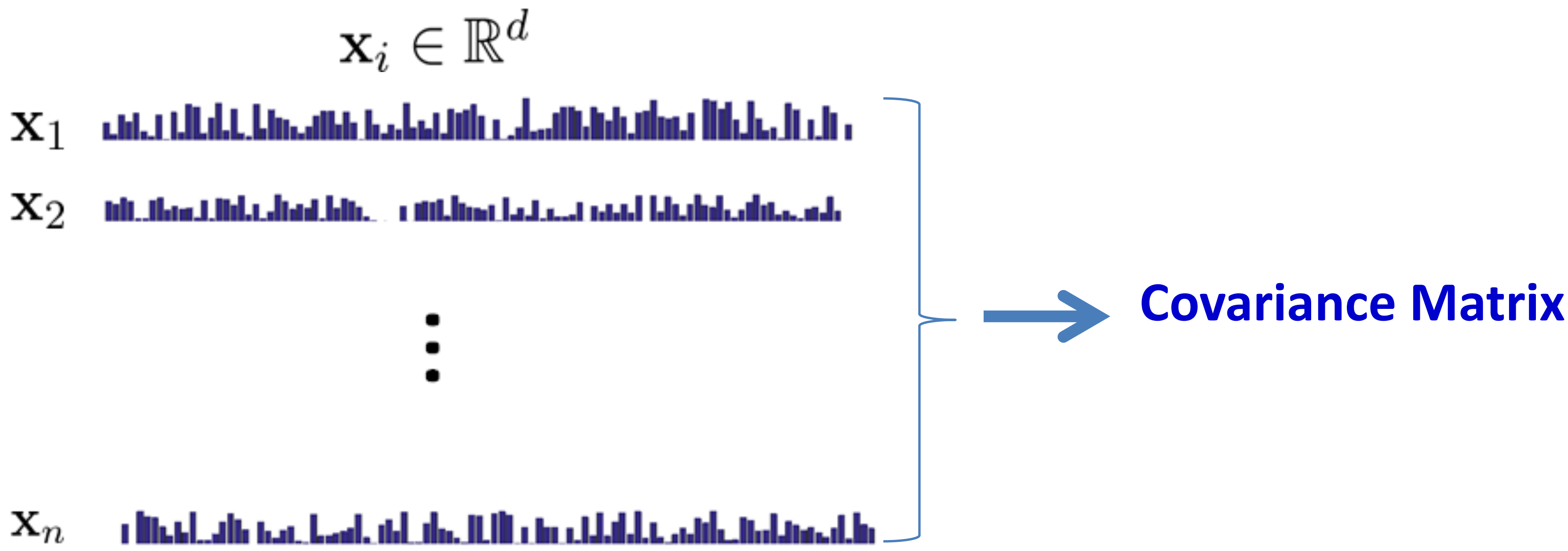
Wang et al., Deep Global Generalized Gaussian Networks, CVPR2019

Brooks et al., Riemannian batch normalization for SPD neural networks, Accepted by NIPS2019

Outline

- Introduction on **Covariance** representation
- Our research work
  - **Discriminatively Learning** Covariance Representation
  - **Exploring Sparse** Inverse Covariance Representation
  - **Moving to Kernel-matrix**-based Representation (KSPD)
  - **Learning KSPD in deep** neural networks
- Conclusion

Motivation



Covariance matrix needs to be **estimated from data**

Motivation

- Covariance estimate becomes **unreliable**
  - High-dimensional (*d*) features
  - Small sample (*n*)

$$\mathrm{rank}(\boldsymbol{\Sigma}_{d\times d}) \leq \min(d, n-1)$$

- Existing work
  - Not consider the **quality** of covariance representation
  - Especially the estimate of **eigenvalues**

Motivation

Stein Kernel

$$k(\boldsymbol{X}, \boldsymbol{Y}) = \exp\left(-\theta \cdot S\left(\boldsymbol{X}, \boldsymbol{Y}\right)\right)$$

$$\text{where} \quad S(\boldsymbol{X}, \boldsymbol{Y}) = \log\left(\det\left(\frac{\boldsymbol{X}+\boldsymbol{Y}}{2}\right)\right) - \frac{1}{2}\log\left(\det(\boldsymbol{X}\boldsymbol{Y})\right)$$

$$\boxed{= \sum_{i=1}^{d} \log \lambda_i \left(\frac{\boldsymbol{X}+\boldsymbol{Y}}{2}\right) - \frac{1}{2}\sum_{i=1}^{d}\left[\log \lambda_i(\boldsymbol{X}) + \log \lambda_i(\boldsymbol{Y})\right]}$$

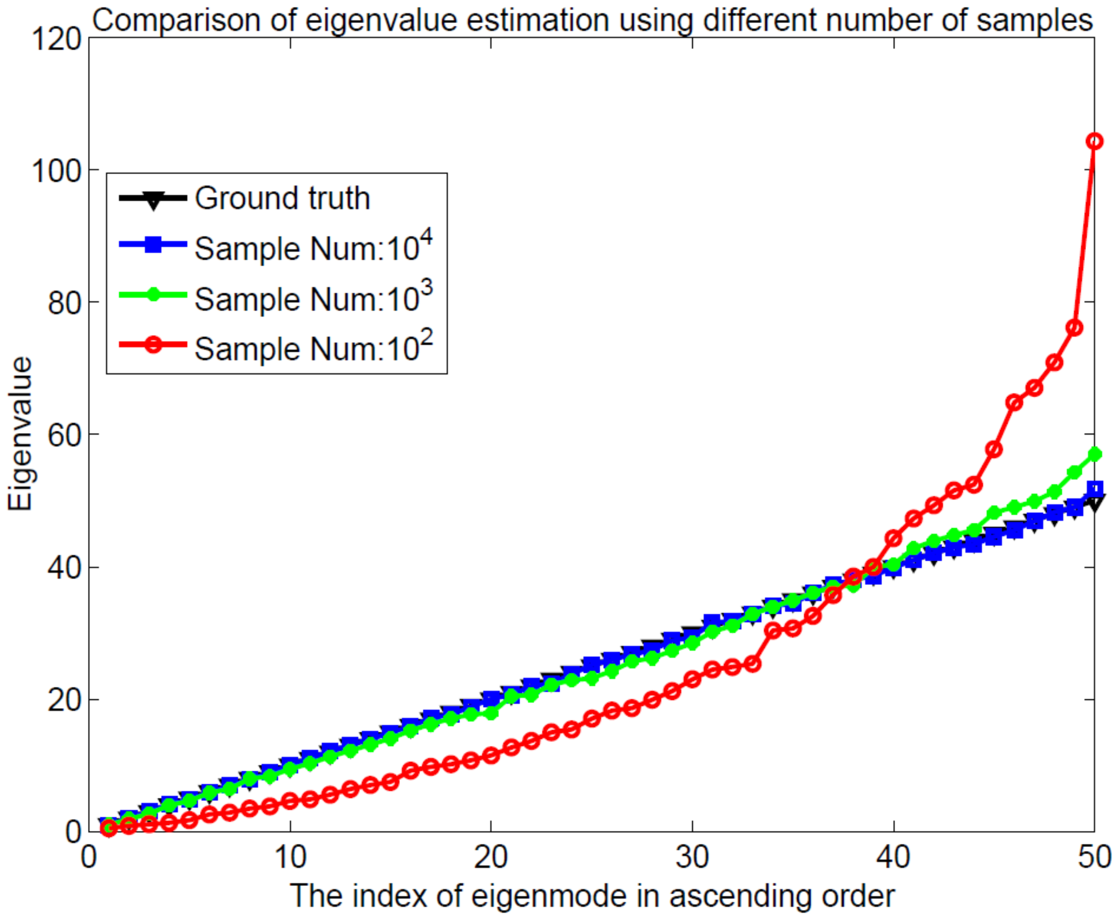
Eigenvalue of $\frac{\boldsymbol{X}+\boldsymbol{Y}}{2}$

Eigenvalue of $\boldsymbol{X}$

Eigenvalue of $\boldsymbol{Y}$

Motivation

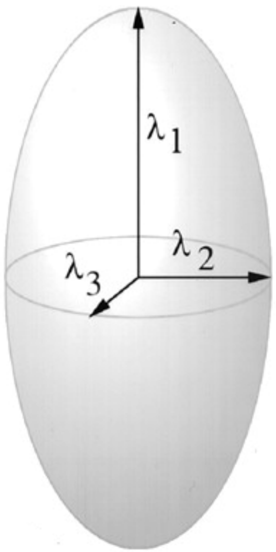
1. Eigenvalue estimation becomes **biased** when the number of samples is **inadequate**

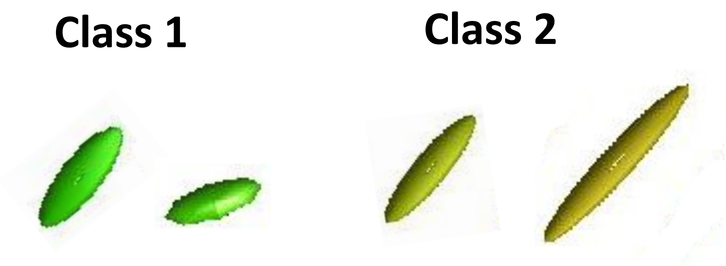


Motivation

2. The **eigenvalues** are **not** collectively manipulated toward greater **discrimination**

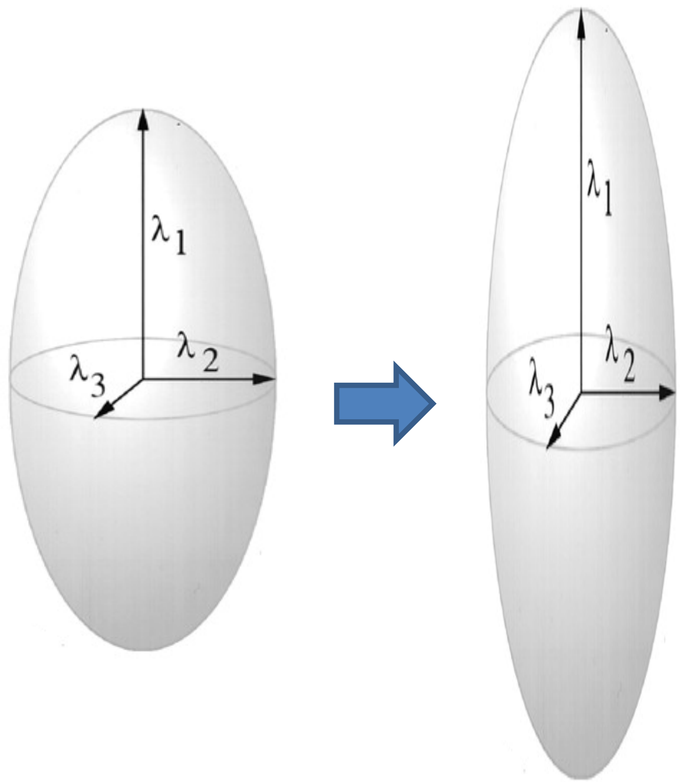
$$\boldsymbol{X} = \lambda_1 \boldsymbol{u}_1 \boldsymbol{u}_1^\top + \lambda_2 \boldsymbol{u}_2 \boldsymbol{u}_2^\top + \cdots + \lambda_d \boldsymbol{u}_d \boldsymbol{u}_d^\top$$

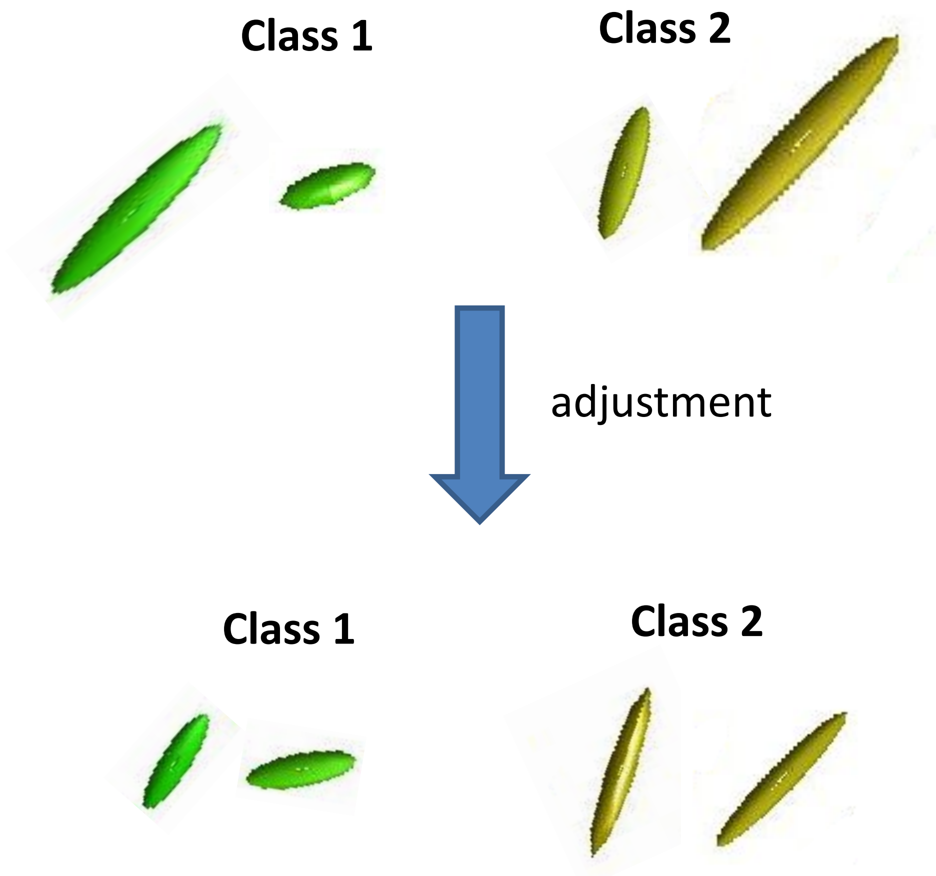




Proposed method

Let’s do a data-dependent “**eigenvalue massage**”





Proposed method

We propose "**Discriminative Covariance Representation**"

$$\boldsymbol{X} = \boldsymbol{U}\boldsymbol{\Lambda}\boldsymbol{U}^\top$$

$$\boldsymbol{\Lambda} = \mathrm{diag}(\lambda_1, \cdots, \lambda_d)$$

$$\tilde{\boldsymbol{X}}_p = \boldsymbol{U} \begin{pmatrix} \lambda_1^{\alpha_1} & & & \\ & \lambda_2^{\alpha_2} & & \\ & & \ddots & \\ & & & \lambda_d^{\alpha_d} \end{pmatrix} \boldsymbol{U}^\top \quad \text{or} \quad \tilde{\boldsymbol{X}}_c = \boldsymbol{U} \begin{pmatrix} \alpha_1\lambda_1 & & & \\ & \alpha_2\lambda_2 & & \\ & & \ddots & \\ & & & \alpha_d\lambda_d \end{pmatrix} \boldsymbol{U}^\top$$

Power-based adjustment

Coefficient-based adjustment

Proposed method

α-**adjusted** S-Divergence:

- **Power-based** adjustment

$$S(\tilde{\boldsymbol{X}}_p, \tilde{\boldsymbol{Y}}_p) = \sum_{i=1}^{d} \log \lambda_i \left(\frac{\tilde{\boldsymbol{X}}_p + \tilde{\boldsymbol{Y}}_p}{2} \right) - \frac{1}{2} \sum_{i=1}^{d} \alpha_i \left(\log \lambda_i(\boldsymbol{X}) + \log \lambda_i(\boldsymbol{Y}) \right)$$

- **Coefficient-based** adjustment

$$S(\tilde{\boldsymbol{X}}_c, \tilde{\boldsymbol{Y}}_c) = \sum_{i=1}^{d} \log \lambda_i \left(\frac{\tilde{\boldsymbol{X}}_c + \tilde{\boldsymbol{Y}}_c}{2} \right) - \frac{1}{2} \sum_{i=1}^{d} \left(2 \log \alpha_i + \log \lambda_i(\boldsymbol{X}) + \log \lambda_i(\boldsymbol{Y}) \right)$$

Discriminative Stein kernel (DSK)

$$k_{\boldsymbol{\alpha}}(\boldsymbol{X}, \boldsymbol{Y}) = \exp\left(-\theta \cdot S_{\boldsymbol{\alpha}}(\boldsymbol{X}, \boldsymbol{Y}) \right)$$

Proposed method

How to **learn** the **optimal** adjustment parameter α?

- **Kernel Alignment** based method
- **Class Separability** based method
- **Radius-margin Bound** based Framework

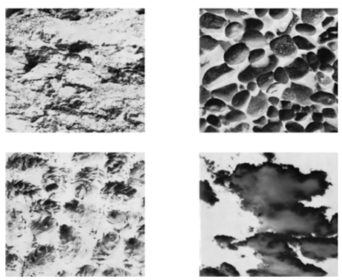
Discriminative Stein kernel (DSK)

$$k_{\boldsymbol{\alpha}}(\boldsymbol{X}, \boldsymbol{Y}) = \exp\left(-\theta \cdot S_{\boldsymbol{\alpha}}\left(\boldsymbol{X}, \boldsymbol{Y}\right)\right)$$

Experimental Result

Data sets

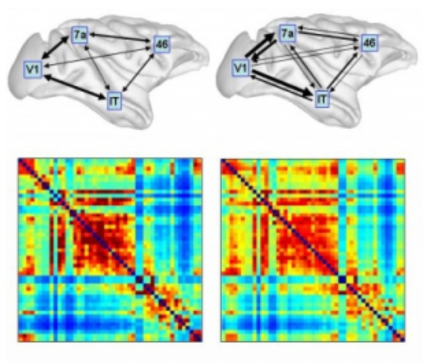
- Brodatz **texture**



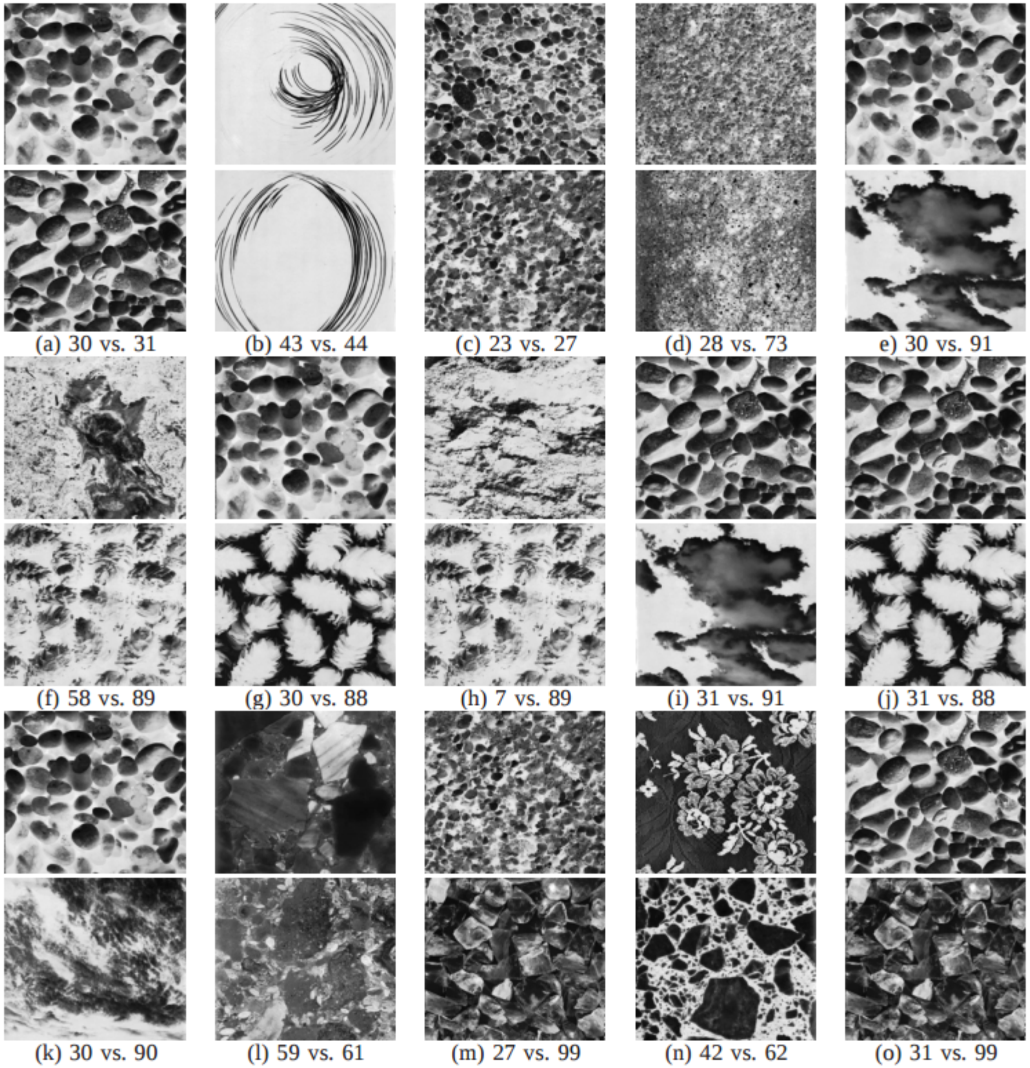
- FERET **face**

- ETH-80 **object**

- ADNI **rs-fMRI**



Experimental Result



The most difficult 15 pairs of Brodatz texture data set

Experimental Result

COMPARISON OF CLASSIFICATION ACCURACY (IN PERCENTAGE) ON EACH OF THE 15 MOST DIFFICULT PAIRS FROM BRODATZ TEXTURE DATA SET

Index	1	2	3	4	5	6	7	8
SK	62.50	67.19	68.75	75.00	75.78	75.79	76.56	77.34
DSK-KA$_p$	**70.31**	**73.44**	**75.00**	**81.25**	**76.56**	**79.69**	**82.81**	**79.69**
Index	9	10	11	12	13	14	15	**Avg.**
SK	78.13	79.69	80.47	81.25	82.04	83.59	85.94	76.67
DSK-KA$_p$	**84.37**	**84.39**	**84.38**	**84.38**	**84.35**	**84.42**	**87.50**	**80.85**

The most difficult 15 pairs of Brodatz texture data set

Discussion

DSK vs. **eigenvalue estimation** improvement methods

Table 1: Comparison of average classification accuracy (in percentage) between DSK and the methods of improving eigenvalue estimation.

Data	n/Dim	sample cov.	[1]	[2]	[3]	DSK
Brodatz	1,024/5 $\approx$ 205	78.01 $\pm$ 0.43	77.50 $\pm$ 0.41	78.00 $\pm$ 0.43	78.00 $\pm$ 0.48	**83.40** $\pm$ 0.58
FERET	98,304/43 $\approx$ 2286	79.70 $\pm$ 3.10	78.10 $\pm$ 2.98	79.70 $\pm$ 3.10	79.68 $\pm$ 3.10	**84.60** $\pm$ 1.71
ETH80	16,384/5 $\approx$ 3276	80.30 $\pm$ 0.79	78.80 $\pm$ 0.89	80.30 $\pm$ 0.82	80.31 $\pm$ 0.59	**82.70** $\pm$ 1.05
fMRI	130/90 $\approx$ 1.44	54.88	54.88	56.10	56.10	**59.76**

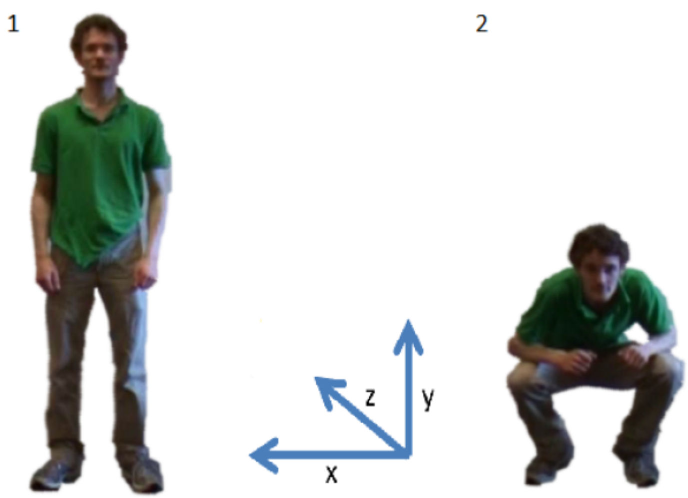
[1] X. Mestre, "Improved estimation of eigenvalues and eigenvectors of covariance matrices using their sample estimates," IEEE Trans. Inf. Theory, vol. 54, pp. 5113–5129, Nov. 2008.
[2] B. Efron and C. Morris, "Multivariate empirical Bayes and estimation of covariance matrices," Ann. Stat., vol. 4, pp. 22–32, 1976.
[3] A. Ben-David and C. E. Davidson, "Eigenvalue estimation of hyper-spectral Wishart covariance matrices from limited number of samples," IEEE Trans. Geosci. Remote Sens., vol. 50, pp. 4384–4396, May 2012.

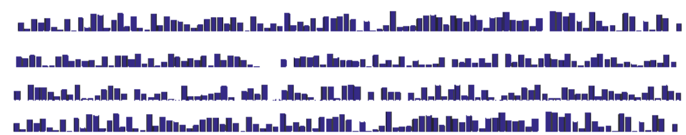
Outline

- Introduction on **Covariance** representation
- Our research work
  - **Discriminatively Learning** Covariance Representation
  - **Exploring Sparse** Inverse Covariance Representation
  - **Moving to Kernel-matrix**-based Representation (KSPD)
  - **Learning KSPD in deep** neural networks
- Conclusion

Introduction

Applications with high dimensions but small sample issue





Small sample 10 ~ 300
High dimensions 50 ~ 400

Introduction

This results in **singular** covariance estimate, which adversely affects representation.

How to address this situation?

Data + **Prior knowledge**

Explore the **underlying structure** of visual features

Proposed SICE representation

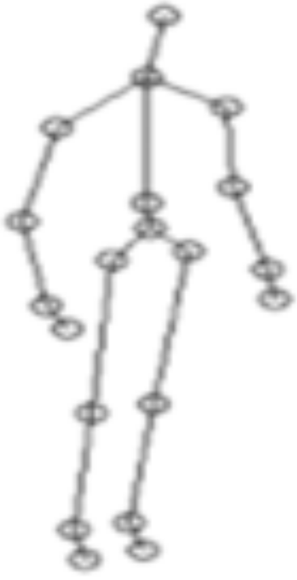
Structure sparsity in skeletal human action recognition

- **Only a small number of joints are directly linked**.

- How to represent such **direct links**?

Sparse Inverse Covariance Estimation

(SICE)



Proposed SICE representation

Assume $\boldsymbol{x}_{d\times 1} \sim \mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma})$

$\boldsymbol{\Sigma}^{-1}_{i,j}$: partial correlation of x_i and x_j (for direct link)

Perform SICE by maximizing penalized log-likelihood

$$\mathbf{S}^* = \arg\max_{\mathbf{S}\succ 0} \left[\log\left(\det(\mathbf{S})\right) - \mathrm{trace}(\mathbf{CS}) - \lambda\|\mathbf{S}\|_1\right]$$

where $\mathbf{C}$ is sample-based covariance matrix

$\|\mathbf{S}\|_1$ imposes the structure sparsity

(Convex, solved by Graphical Lasso, 0.014 CPU second for $\mathbf{S}_{100\times 100}$)

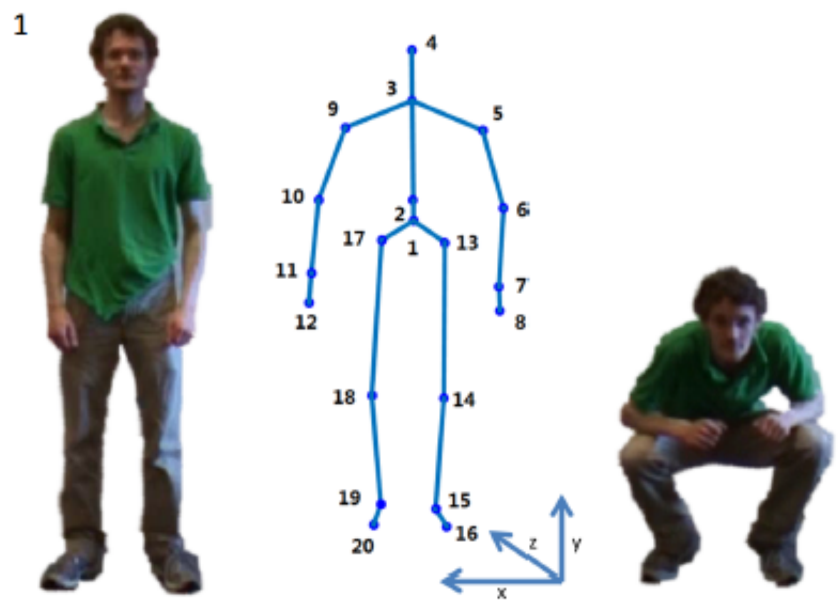
Proposed SICE representation

Properties of SICE representation:

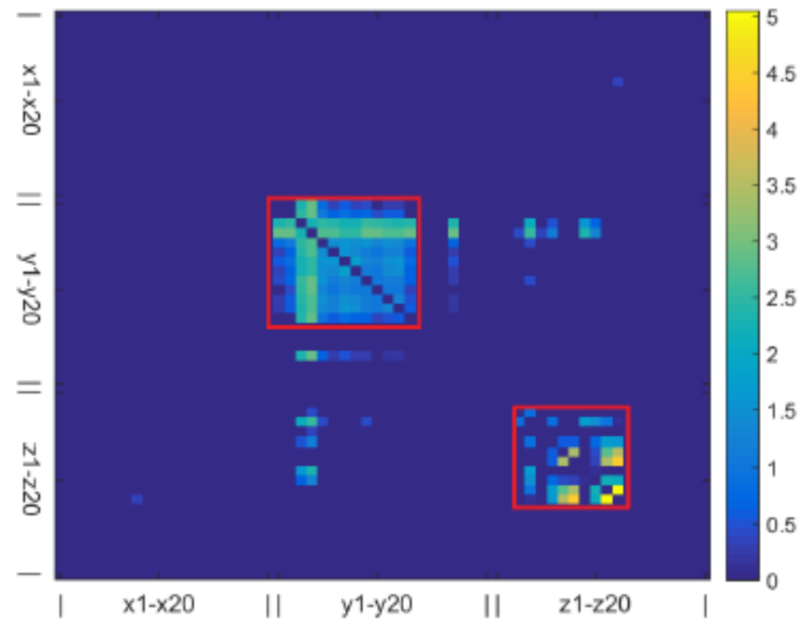
- is guaranteed to be **nonsingular**
- reduces over-fitting, giving **more reliable** representation
- Measures the **partial correlation**, allowing the **sparsity prior** to be conveniently imposed

$$\mathbf{S}^* = \arg\max_{\mathbf{S}\succ 0} \left[\log\left(\det(\mathbf{S})\right) - \operatorname{trace}(\mathbf{CS}) - \lambda\|\mathbf{S}\|_1\right]$$

Application to Skeletal Action Recognition



(a) "Crouch or hide" action from MSRC-12 data set.



(b) Proposed SICE-RP

Application to Skeletal Action Recognition

Table 1: Comparison on HDM05 data set (Two experiments).

Methods in comparison	14 classes Accuracy	All classes Accuracy
Cov-$J_{\mathcal{H}}$-SVM	82.5	Not reported
RSR	76.1	Not reported
RSR-ML	81.9	40.0
CDL	79.8	Not reported
Cov-RP	91.5	58.9
InverseCov-RP	91.5	58.9
SICE-RP (proposed)	**96.8**	**67.6**

Table 1: Comparison on MSR-DailyActivity3D data set.

Methods in comparison	Accuracy
Moving Pose	73.8
Local HON4D	80.0
Actionlet Ensemble	86.0
SNV	86.3
Cov-$J_{\mathcal{H}}$-SVM	75.0
Cov-RP	85.0
InverseCov-RP	85.0
SICE-RP (proposed)	**93.1**

Table 2: Comparison on MSRC-12 data set.

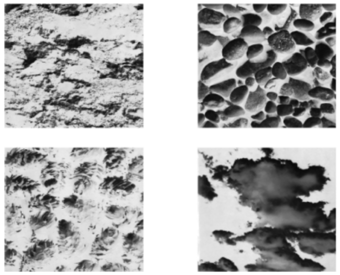
Methods in comparison	Accuracy
Cov-$J_{\mathcal{H}}$-SVM	89.8
Hierarchy of Cov3DJs	91.7
Cov-RP	89.2
InverseCov-RP	89.2
SICE-RP (proposed)	**92.5**

Application to other tasks

The principle of ``**Bet on sparsity**''

Table 1: Comparison of classification performance on object classification data sets.

Methods	Brodatz (texture)	FERET (face)	ETH80 (object)
Cov-RP	81.2	81.0	94.0
InverseCov-RP	81.2	81.0	94.0
SICE-RP (proposed)	**81.5**	**83.1**	**94.1**



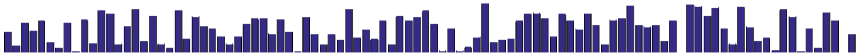
Outline

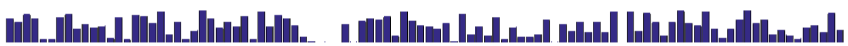
- Introduction on **Covariance** representation
- Our research work
  - **Discriminatively Learning** Covariance Representation
  - **Exploring Sparse** Inverse Covariance Representation
  - **Moving to Kernel-matrix**-based Representation (KSPD)
  - **Learning KSPD in deep** neural networks
- Conclusion

Introduction

Again, look into **Covariance representation**

$$\boldsymbol{\Sigma} = \frac{1}{n-1} \sum_{i=1}^{n} (\boldsymbol{x}_i - \boldsymbol{\mu})(\boldsymbol{x}_i - \boldsymbol{\mu})^\top$$

$\mathbf{x}_1$ 

$\mathbf{x}_2$ 

$$\vdots$$

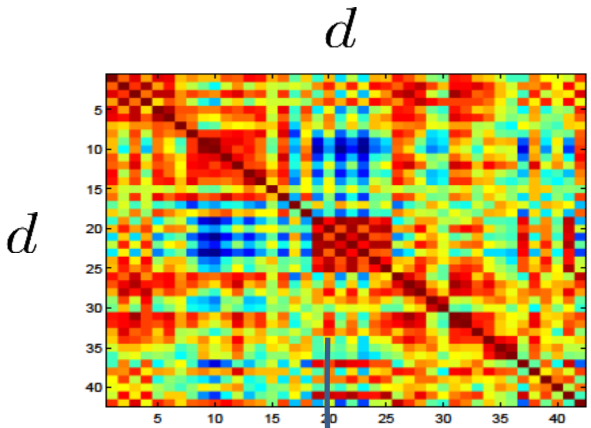
$\mathbf{x}_n$ 

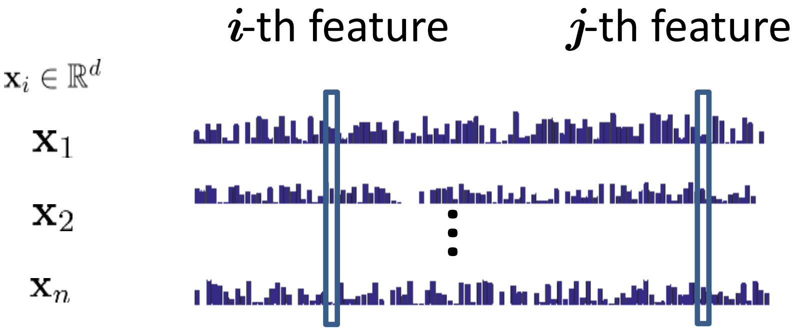
$$\mathbf{x}_i \in \mathbb{R}^d$$

Introduction

Again, look into **Covariance representation**

$$\boldsymbol{\Sigma} = \frac{1}{n-1}\sum_{i=1}^{n}(\boldsymbol{x}_i - \boldsymbol{\mu})(\boldsymbol{x}_i - \boldsymbol{\mu})^\top$$





$$\mathbf{C}_{ij} = \left\langle \frac{\bar{\mathbf{f}}_i}{\sqrt{n-1}}, \frac{\bar{\mathbf{f}}_j}{\sqrt{n-1}} \right\rangle$$

Just a linear kernel function!

Introduction

Covariance representation

$$\mathbf{C}_{ij} = \left\langle \frac{\bar{\mathbf{f}}_i}{\sqrt{n-1}}, \frac{\bar{\mathbf{f}}_j}{\sqrt{n-1}} \right\rangle$$

Resulting issues:

- Only modeling **linear** correlation of features.
- A single, **fixed** representation form.
- **Unreliable** or even **singular** covariance estimate.

Proposed kernel-matrix representation

Let's use a **kernel matrix** instead

$$\mathbf{C}_{ij} = \left\langle \frac{\bar{\mathbf{f}}_i}{\sqrt{n-1}}, \frac{\bar{\mathbf{f}}_j}{\sqrt{n-1}} \right\rangle$$

Covariance

$$\Downarrow$$

$$\mathbf{M}_{ij} = \langle \phi(\mathbf{f}_i), \phi(\mathbf{f}_j) \rangle = \kappa(\mathbf{f}_i, \mathbf{f}_j)$$

SPD Matrix!

Advantages:

- Model **nonlinear relationship** between features;
- For many kernels, **M** is **guaranteed to be nonsingular**, no matter what the feature dimensions and sample size are.
- **Maintain the size** of covariance representation and the computational load.

Application to Skeletal Action Recognition

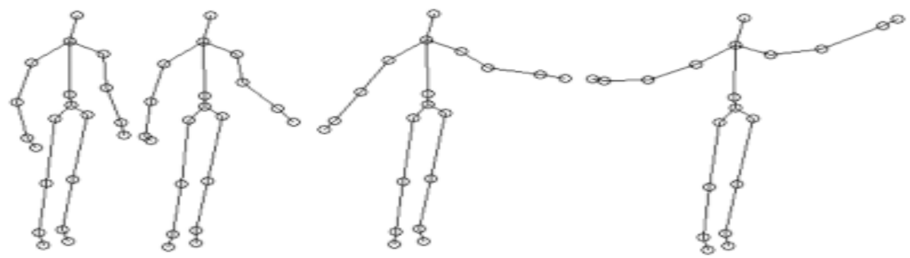
Comparison on MSR-Action3D data set.

Methods in comparison	Accuracy
Pose Set [25]	90.0
Hierarchy of Cov3DJs [10]	90.5
Moving Pose [31]	91.7
Lie Group [24]	92.5
SNV [29]	93.1
Spatiotemp. Features Fusing [32]	94.3
Cov-RP [22]	74.0
Cov-$J_{\mathcal{H}}$-SVM [7]	80.4
Ker-RP-POL (proposed)	96.2
Ker-RP-RBF (proposed)	**96.9**

Comparison on MSR-DailyActivity3D data set.

Methods in comparison	Accuracy
Moving Pose [31]	73.8
Local HON4D [13]	80.0
Actionlet Ensemble [26]	86.0
SNV [29]	86.3
Cov-RP [22]	85.0
Cov-$J_{\mathcal{H}}$-SVM [7]	75.0
Ker-RP-POL (proposed)	**96.9**
Ker-RP-RBF (proposed)	96.3

Application to Skeletal Action Recognition



Comparison on HDM05 data set (Two experiments).

Methods in comparison	14 classes Accuracy	All classes Accuracy
CDL [27]	79.8	Not reported
RSR [8]	76.1	Not reported
RSR-ML [6]	81.9	40.0
Cov-RP [22]	91.5	58.9
Cov-$J_{\mathcal{H}}$-SVM [7]	82.5	-
Ker-RP-POL (proposed)	93.6	64.3
Ker-RP-RBF (proposed)	**96.8**	**66.2**

$\star$The result of Cov-$J_{\mathcal{H}}$-SVM [7] is not obtained in 35 hours.

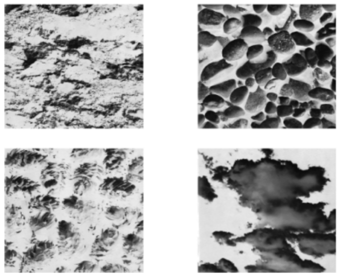
Comparison on MSRC-12 data set.

Methods in comparison	Accuracy
Hierarchy of Cov3DJs [10]	91.7
Cov-RP [22]	89.2
Cov-$J_{\mathcal{H}}$-SVM [7]	89.2
Ker-RP-POL (proposed)	90.5
Ker-RP-RBF (proposed)	**92.3**

Application to Object Recognition

Comparison on object classification data sets.

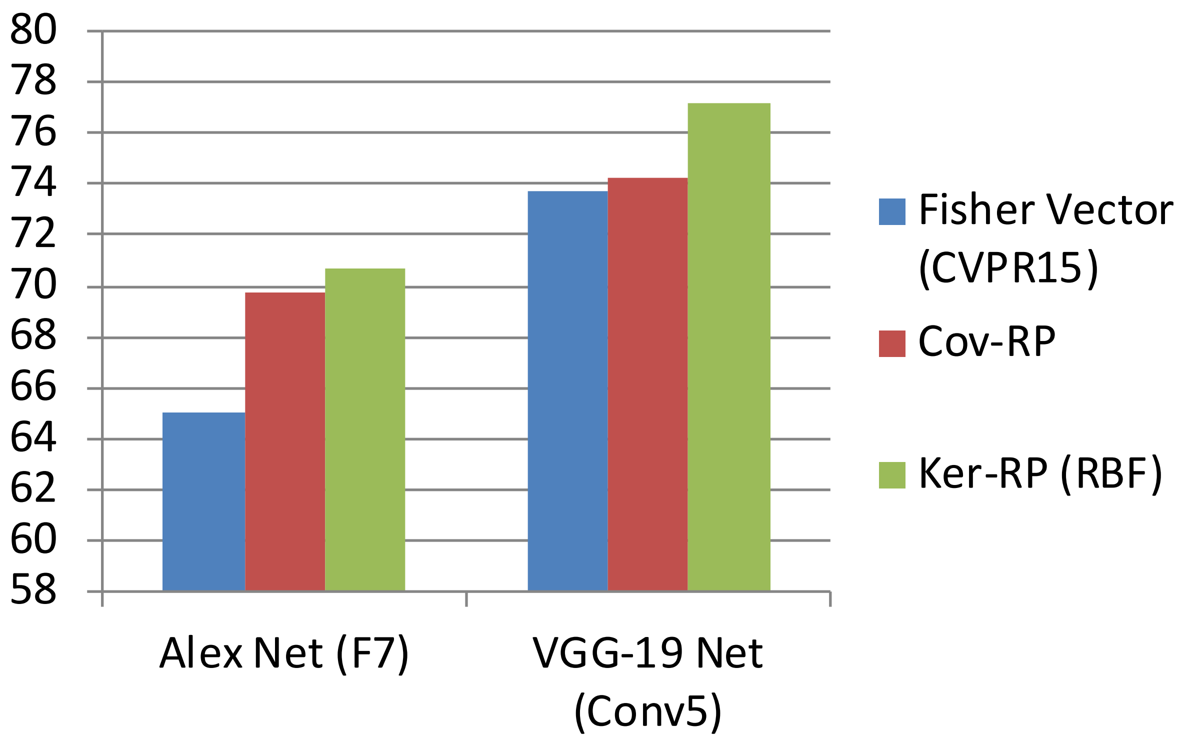
Methods	Brodatz (texture)	FERET (face)	ETH80 (object)
Cov-RP [22]	81.2	81.0	94.0
Ker-RP-POL (proposed)	77.9	82.4	93.8
Ker-RP-RBF (proposed)	**84.9**	**85.4**	**94.8**



Application to Deep Learning Features

Comparison on MIT Indoor Scenes Data Set

(Classification accuracy in percentage)



Discussion

SICE vs. Kernel matrix: which is better?

Table 1: Comparison between SICE-RP and Kernel representation.

Data set	Cov-RP	SICE-RP	Ker-RP-RBF
MSRC-12	89.2	**92.5**	92.3
HDM05 (14 classes)	91.5	**96.8**	**96.8**
HDM05 (100 classes)	58.9	**67.6**	66.2
MSR-Action3D	74.0	96.5	**96.9**
MSR-DailyActivity3D	85.0	93.1	**96.3**
Brodatz	81.2	81.5	**84.9**
FERET	81.0	83.1	**85.4**
ETH80	94.0	94.1	**94.8**

Discussion

SICE vs. Kernel matrix representation: which is better?

Table 1: Comparison between SICE and Kernel representation.

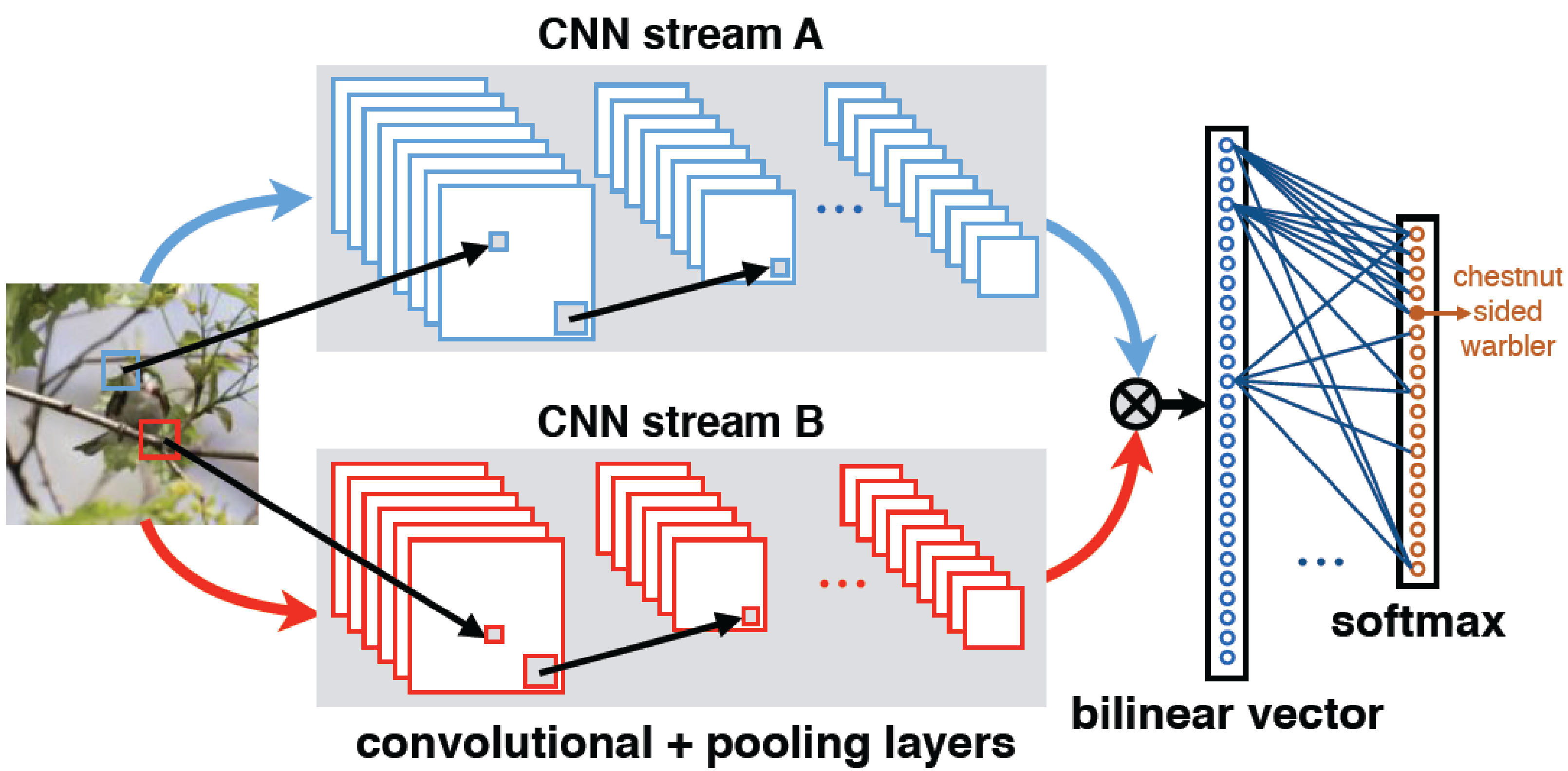
Criterion	Cov-RP	SICE-RP	Ker-RP
Robust to small sample & high dimensionality	×	√	√
Prior knowledge incorporation	×	√	√
Guaranteed to be SPD	×	√	√
Linear technique	√	√	×
Flexibility	×	×	√
Free of parameter tuning	√	×	×

Outline

- Introduction on **Covariance** representation
- Our research work
  - **Discriminatively Learning** Covariance Representation
  - **Exploring Sparse** Inverse Covariance Representation
  - **Moving to Kernel-matrix**-based Representation (KSPD)
  - **Learning KSPD in deep** neural networks
- Conclusion

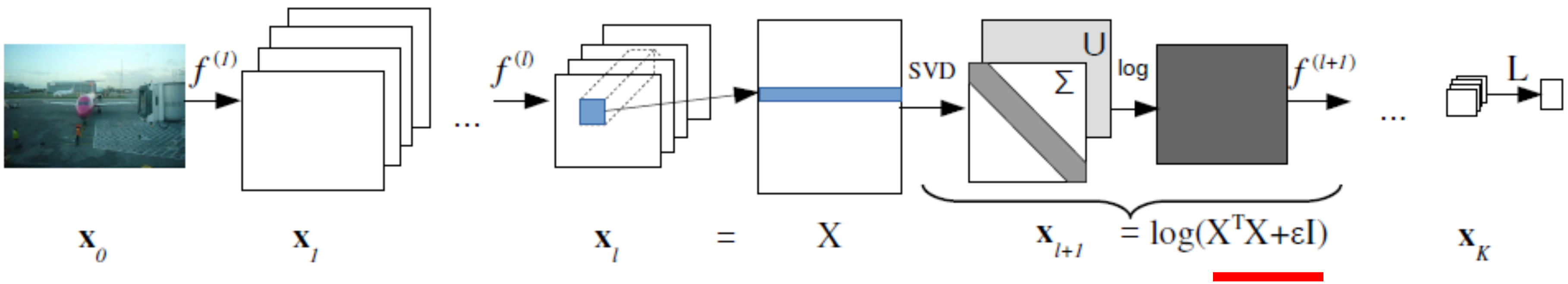
Covariance representation

Integration with Deep Learning



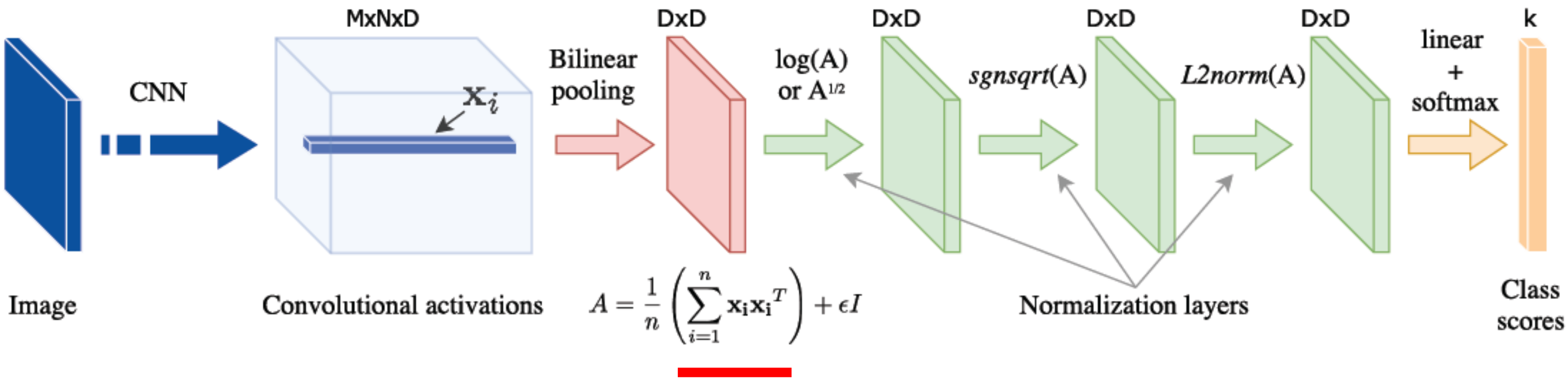
Bilinear CNN Models for Fine-grained Visual Recognition, Lin et al, ICCV2015

Integration with Deep Learning



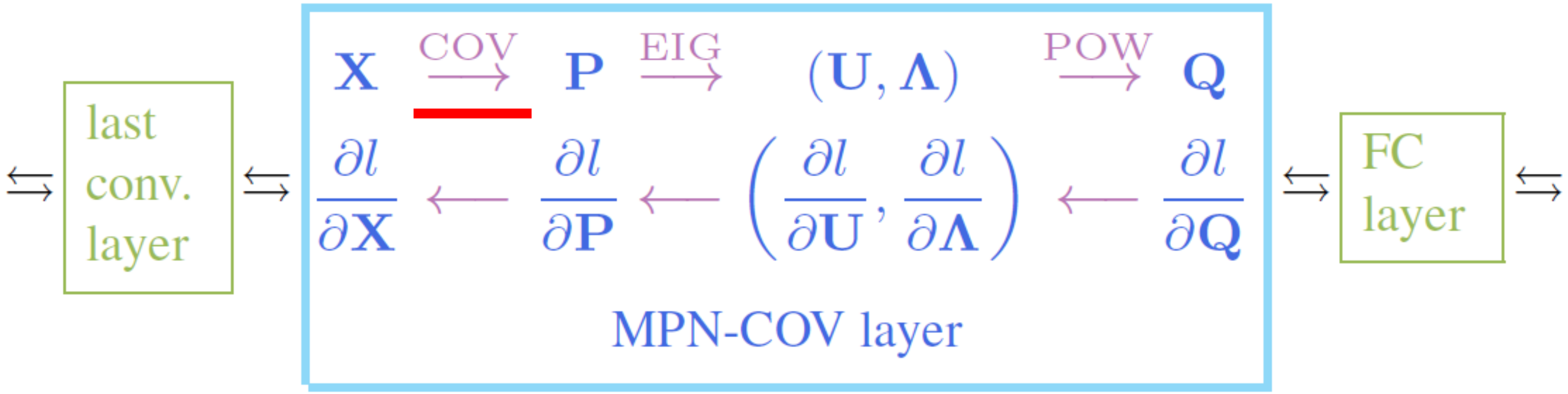
Matrix Backpropagation for Deep Networks with Structured Layers,
Ionescu et al, ICCV2015

Integration with Deep Learning



Improved Bilinear Pooling with CNN, Lin and Maji, BMVC2017

Integration with Deep Learning



Is Second-order Information Helpful for Large-scale Visual Recognition?,
Li et al., ICCV2017

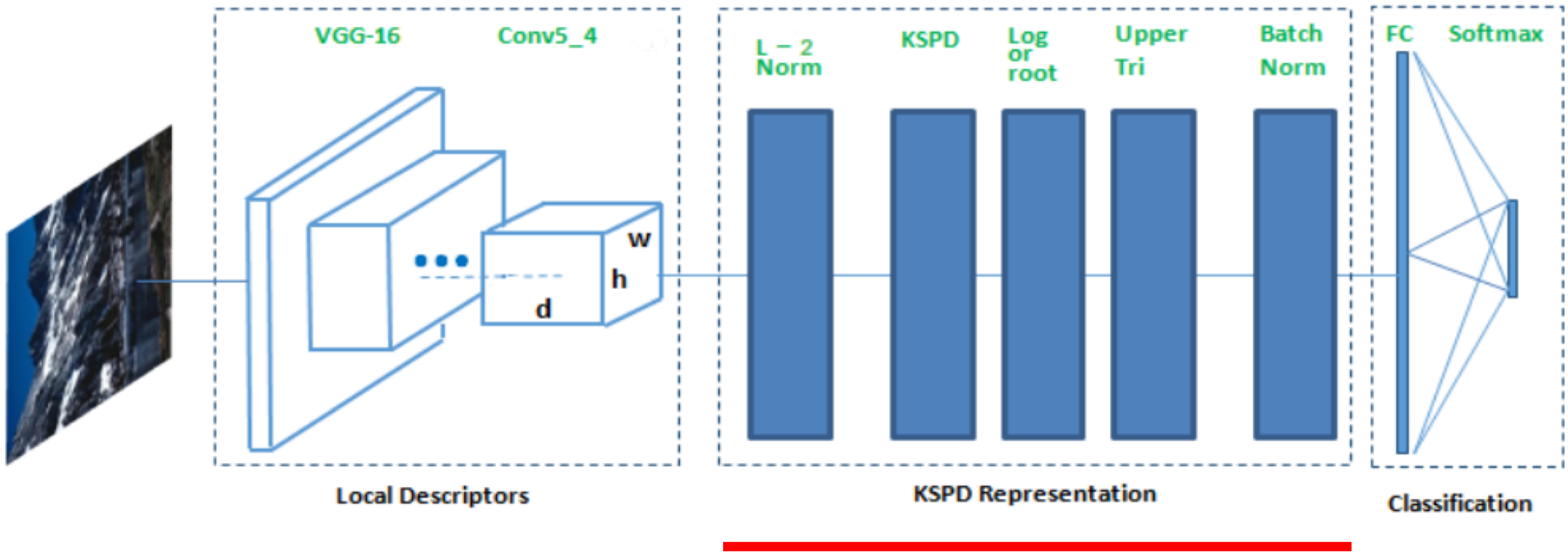
Proposed DeepKSPD

Motivation

- The **kernel-matrix-based SPD representation**
  - has **not** been developed upon **deep** local descriptors
  - has **not** been jointly learned via **deep** learning
- Existing **matrix backpropagation** for learning covariance-representation via deep networks
  - encounters **numerical stability issue**

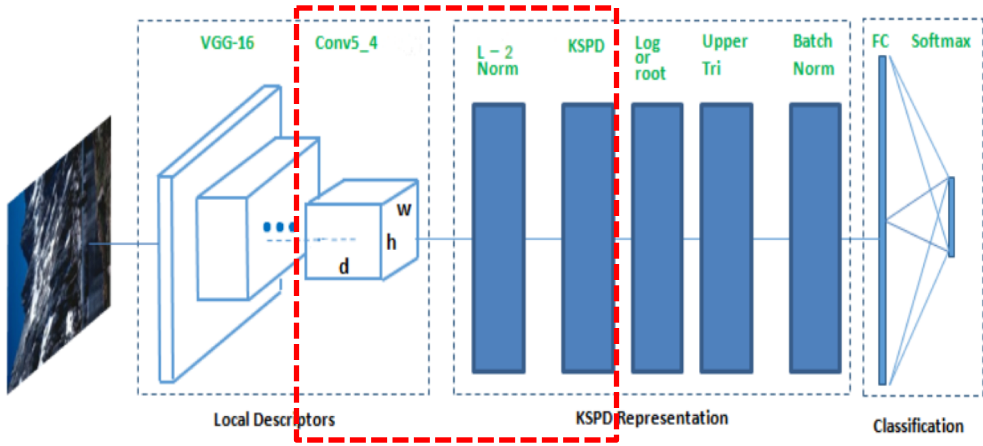
Proposed DeepKSPD

Architecture and layers



Proposed DeepKSPD

Matrix backpropagation



$$\boxed{\boldsymbol{X}_{d\times n}} \to \overset{\boldsymbol{A}_{d\times d}}{\boxed{\boldsymbol{X}\boldsymbol{X}^T}} \to \overset{\boldsymbol{E}_{d\times d}}{\boxed{(\boldsymbol{I}\circ\boldsymbol{A})\mathbf{1}+\mathbf{1}^T(\boldsymbol{I}\circ\boldsymbol{A})^T-2\boldsymbol{A}}} \to \overset{\boldsymbol{K}_{d\times d}}{\boxed{\exp[-\theta\cdot\boldsymbol{E}]}} \to \boxed{\cdots} \to \boxed{J}$$

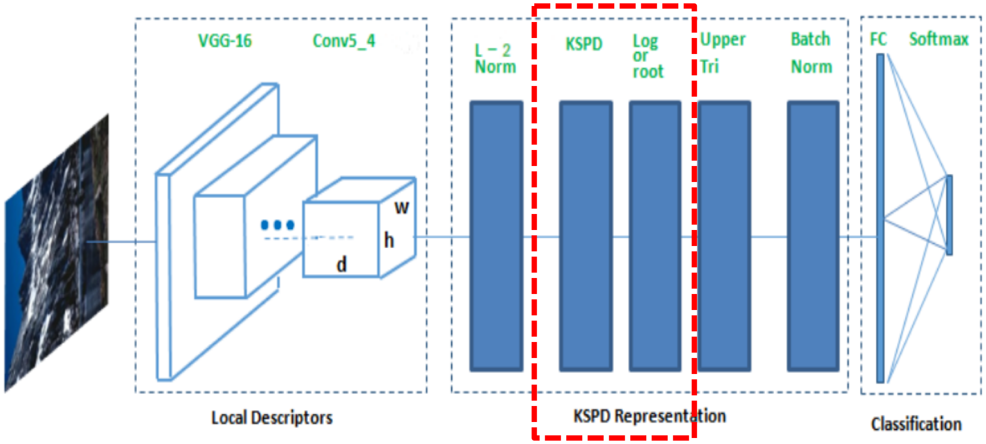
$$\boldsymbol{K} = \exp\left[-\theta\cdot\left((\boldsymbol{I}\circ\boldsymbol{X}\boldsymbol{X}^T)\mathbf{1}+\mathbf{1}^T(\boldsymbol{I}\circ\boldsymbol{X}\boldsymbol{X}^T)^T-2\boldsymbol{X}\boldsymbol{X}^T\right)\right]$$

$$\frac{\partial J}{\partial \boldsymbol{X}} = \left(\frac{\partial J_1}{\partial \boldsymbol{A}}+\left(\frac{\partial J_1}{\partial \boldsymbol{A}}\right)^T\right)\boldsymbol{X} \qquad \frac{\partial J_1}{\partial \boldsymbol{A}} = \boldsymbol{I}\circ\left(\left(\frac{\partial J_2}{\partial \boldsymbol{E}}+\left(\frac{\partial J_2}{\partial \boldsymbol{E}}\right)^T\right)\mathbf{1}^T\right)-2\frac{\partial J_2}{\partial \boldsymbol{E}}$$

$$\frac{\partial J_2}{\partial \boldsymbol{E}} = (-\theta\boldsymbol{K})\circ\boxed{\frac{\partial J_3}{\partial \boldsymbol{K}}} \qquad \frac{\partial J}{\partial \theta} = \mathrm{trace}\left(\left(\frac{\partial J_3}{\partial \boldsymbol{K}}\right)^T(-\boldsymbol{K}\circ\boldsymbol{E})\right)$$

Proposed DeepKSPD

Matrix backpropagation



H* = f(*K*) on the kernel matrix *K

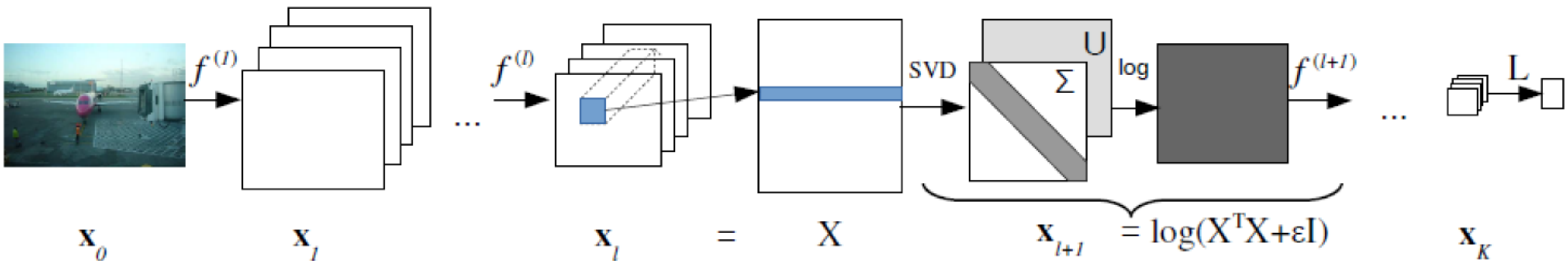
$$\boldsymbol{K} = \boldsymbol{U}\boldsymbol{D}\boldsymbol{U}^T \qquad \boldsymbol{H} = \boldsymbol{U}f(\boldsymbol{D})\boldsymbol{U}^T$$

$$J(\boldsymbol{X}) = J_4(\boldsymbol{H}) = J_4(f(\boldsymbol{K})).$$

$$\frac{\partial J_3}{\partial \boldsymbol{K}} \sim \frac{\partial J_4}{\partial \boldsymbol{H}} \quad ?$$

Proposed DeepKSPD

Existing matrix backpropagation



Matrix Backpropagation for Deep Networks with Structured Layers, Ionescu et al, ICCV2015

$$\boxed{\frac{\partial J_3}{\partial \boldsymbol{K}}} = \boldsymbol{U}\left\{\left(\tilde{\boldsymbol{G}} \circ \left(2\boldsymbol{U}^T \left(\boxed{\frac{\partial J_4}{\partial \boldsymbol{H}}}\right)_{sym} \boldsymbol{U}\log(\boldsymbol{D})\right)\right) + \left(\boldsymbol{D}^{-1}\left(\boldsymbol{U}^T \boxed{\frac{\partial J_4}{\partial \boldsymbol{H}}} \boldsymbol{U}\right)\right)_{diag}\right\}\boldsymbol{U}^T, \quad (16)$$

where $\boldsymbol{K} = \boldsymbol{U}\boldsymbol{D}\boldsymbol{U}^T$; $\boxed{\tilde{g}_{ij} = (\lambda_i - \lambda_j)^{-1}}$ *when* $i \neq j$ *and zero otherwise;* $\boldsymbol{A}_{diag}$ *means the off-diagonal entries of* $\boldsymbol{A}$ *are all set to zeros; and* $\boldsymbol{A}_{sym}$ *is defined to represent* $(\boldsymbol{A} + \boldsymbol{A}^T)/2$.

Proposed DeepKSPD

Result from the literature of Operator Theory (1951)

Theorem 1 (pp.60, [20]) *Let* $\mathbb{M}_d$ *be the set of* $d \times d$ *real symmetric matrices. Let* I *be an open interval and* $\mathbb{M}_d(I)$ *is the set of all real symmetric matrices whose eigenvalues belong to* I*. Let* $C^1(I)$ *be the space of continuously differentiable real functions on* I*. Every function* f *in* $C^1(I)$ *induces a differentiable map from* $\boldsymbol{A}$ *in* $\mathbb{M}_d(I)$ *to* $f(\boldsymbol{A})$ *in* $\mathbb{M}_d$*. Let* $Df_{\boldsymbol{A}}(\cdot)$ *denote the derivative of* $f(\boldsymbol{A})$ *at* $\boldsymbol{A}$*. It is a linear map from* $\mathbb{M}_d$ *to itself. When applied to* $\boldsymbol{B} \in \mathbb{M}_d$*,* $Df_{\boldsymbol{A}}(\cdot)$ *is given by the Daleckiĭ-Kreĭn formula as*

$$\frac{\partial J_4}{\partial \boldsymbol{H}}$$

$$\frac{\partial J_3}{\partial \boldsymbol{K}} \qquad Df_{\boldsymbol{A}}(\boldsymbol{B}) = \boldsymbol{U}\left(\boldsymbol{G} \circ \left(\boldsymbol{U}^T \boldsymbol{B} \boldsymbol{U}\right)\right)\boldsymbol{U}^T, \tag{11}$$

where $\boldsymbol{A} = \boldsymbol{U}\boldsymbol{D}\boldsymbol{U}^T$ *is the eigen-decomposition of* $\boldsymbol{A}$ *with* $\boldsymbol{D} = \mathrm{diag}(\lambda_1, \cdots, \lambda_d)$*, and* $\circ$ *is the entry-wise product. The entry of the matrix* $\boldsymbol{G}$ *is defined as*

$$\boxed{g_{ij} = \begin{cases} \frac{f(\lambda_i) - f(\lambda_j)}{\lambda_i - \lambda_j} & \text{if } \lambda_i \neq \lambda_j \\ f'(\lambda_i), & \text{otherwise.} \end{cases}} \tag{12}$$

20. Bhatia, R.: Positive Definite Matrices. Princeton University Press (2015)

Proposed DeepKSPD

Existing matrix backpropagation (Ionescu et al, ICCV2015)

$$\frac{\partial J_3}{\partial \boldsymbol{K}} = \boldsymbol{U}\left\{\left(\tilde{\boldsymbol{G}} \circ \left(2\boldsymbol{U}^T\left(\frac{\partial J_4}{\partial \boldsymbol{H}}\right)_{sym}\boldsymbol{U}\log(\boldsymbol{D})\right)\right) + \left(\boldsymbol{D}^{-1}\left(\boldsymbol{U}^T\frac{\partial J_4}{\partial \boldsymbol{H}}\boldsymbol{U}\right)\right)_{diag}\right\}\boldsymbol{U}^T, \tag{16}$$

where $\boldsymbol{K} = \boldsymbol{U}\boldsymbol{D}\boldsymbol{U}^T$; $\tilde{g}_{ij} = (\lambda_i - \lambda_j)^{-1}$ *when* $i \neq j$ *and zero otherwise;* $\boldsymbol{A}_{diag}$ *means the off-diagonal entries of* $\boldsymbol{A}$ *are all set to zeros; and* $\boldsymbol{A}_{sym}$ *is defined to represent* $(\boldsymbol{A} + \boldsymbol{A}^T)/2$.

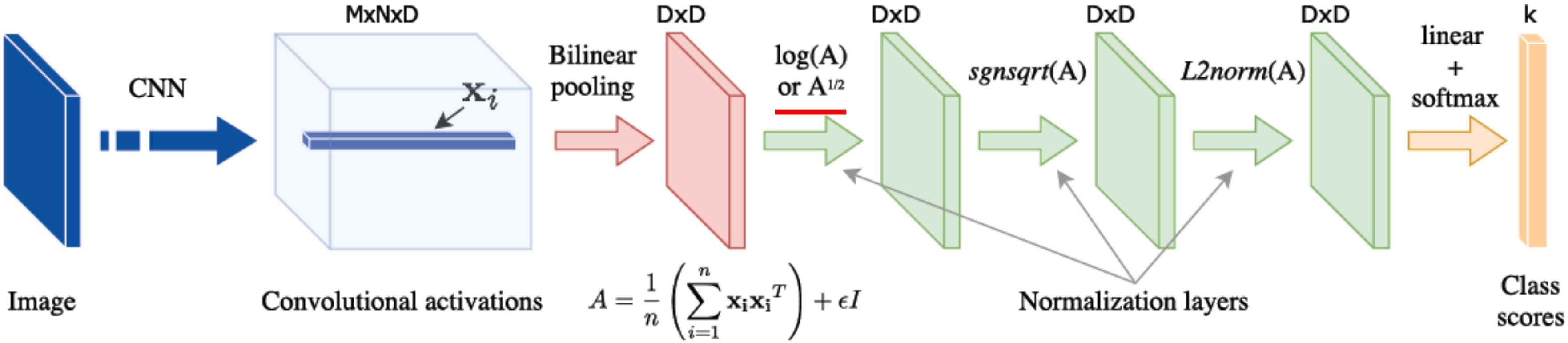
Proposed matrix backpropagation

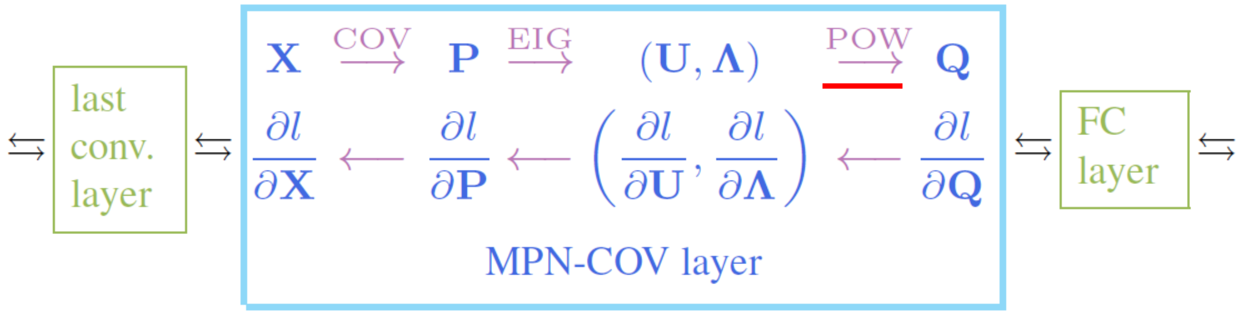
$$\frac{\partial J_3}{\partial \boldsymbol{K}} = \boldsymbol{U}\left(\boldsymbol{G} \circ \left(\boldsymbol{U}^T\frac{\partial J_4}{\partial \boldsymbol{H}}\boldsymbol{U}\right)\right)\boldsymbol{U}^T$$

$$g_{ij} = \begin{cases} \frac{f(\lambda_i)-f(\lambda_j)}{\lambda_i-\lambda_j} & \textit{if } \lambda_i \neq \lambda_j \\ f'(\lambda_i), & \textit{otherwise.} \end{cases}$$

What is their relationship?

Generalise to matrix α-rooting normalisation

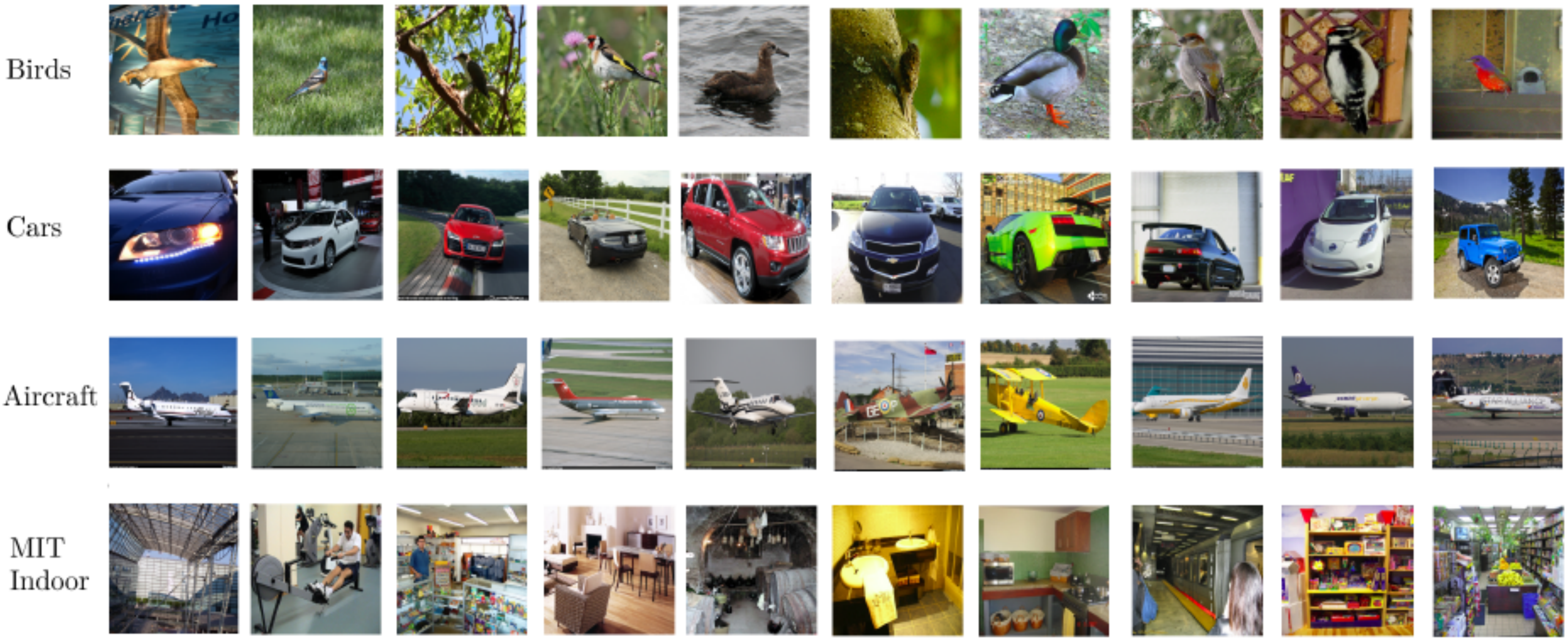




$$f(\lambda) = \lambda^{\alpha} \quad \Longrightarrow \quad \frac{\partial J}{\partial \alpha} = \text{trace}\left(\left(\frac{\partial \boldsymbol{J}_4}{\partial \boldsymbol{H}}\right)^T \left[\boldsymbol{U}(\log(\boldsymbol{D}) \circ \boldsymbol{D}^{\alpha})\boldsymbol{U}^T\right]\right)$$

Experimental Result

Fine-grained Image Recognition



Experimental Result

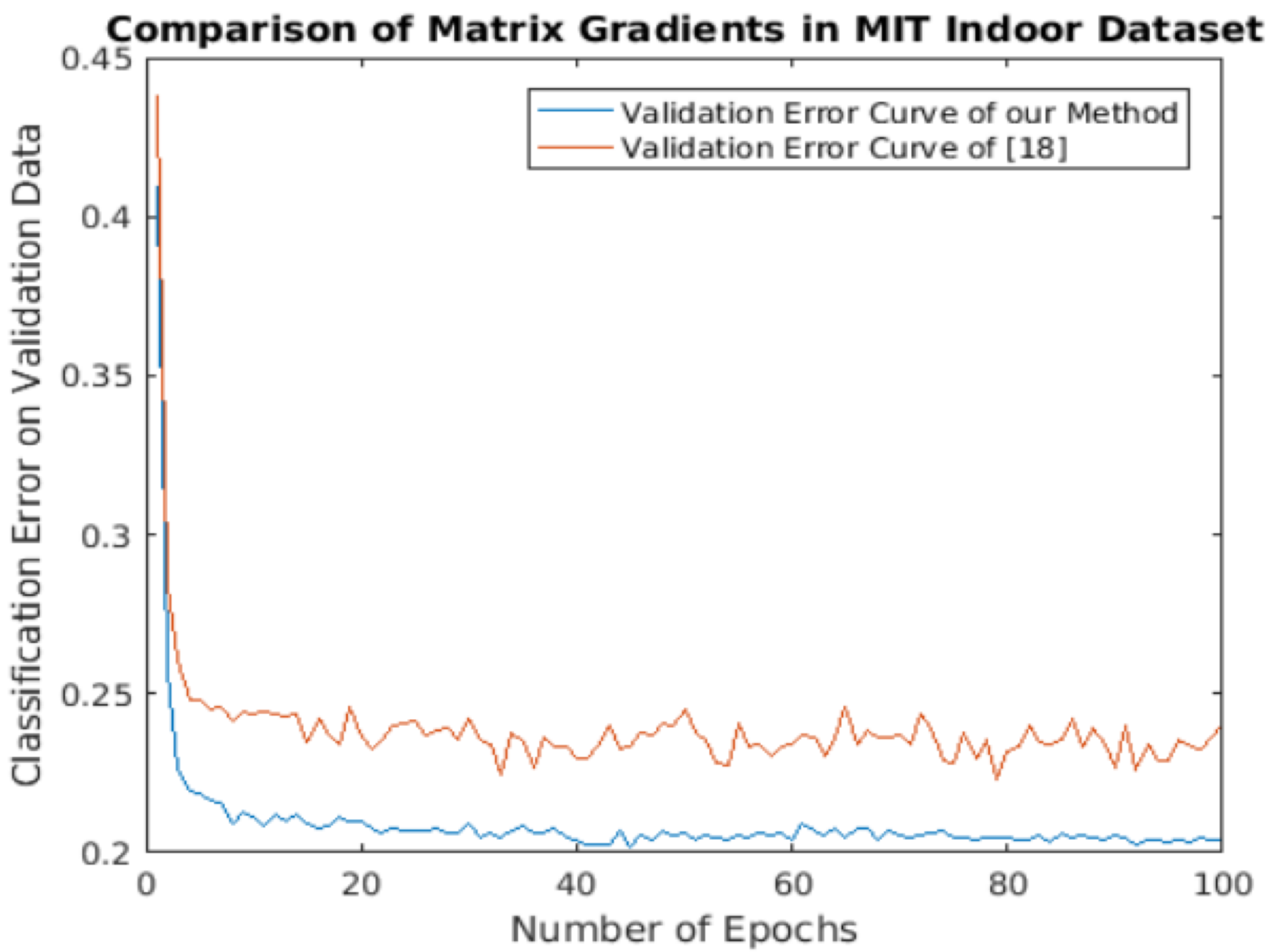
Fine-grained Image Recognition

Table 1. Comparison of Methods

ACC (%)	MIT indoor	Cars	Aircraft	Birds	Average
Symbiotic Model [29]	–	78.0	72.5	–	–
FV-revisit [30]	–	82.7	80.7	–	–
FV-SIFT [27]	–	59.2	61.0	18.8	–
FC-VGG [21]	67.6	36.5	45.0	61.0	52.5
FV-VGG [28]	73.7	75.2	72.7	71.3	73.1
FV-VGG-ft [21]	–	85.7	78.7	74.7	73.1
COV-VGG	74.2	80.3	81.4	76	78.0
KSPD-VGG (**proposed**)	77.2	83.5	83.8	78.5	80.1
BCNN [13]	77.6	91.3	86.6	84.1	84.5
Improved BCNN [12]	–	92.0	88.5	85.8	–
CBP [14]	76.17	–	–	84.0	–
LRBP [11]	–	90.9	87.3	84.2	–
KP [17]	–	92.4	86.9	86.2	–
DeepKSPD-logm (**proposed**)	79.6	90.5	**91.5**	84.8	86.6
DeepKSPD-rootm (**proposed**)	**81.0**	**93.2**	91.0	**86.5**	**87.9**

Experimental Result

Numerical stability of backpropagation



Experimental Result

DeepKSPD vs DeepCOV

ACC (%)	MIT indoor	Cars	Aircraft	Birds
Improved BCNN [12]	–	92.0	88.5	85.8
DeepCOV-rootm	79.2	91.7	88.7	85.4
DeepKSPD-rootm	**81.0**	**93.2**	**91.0**	**86.5**

Experimental Result

Ablation study

- **Learning width θ in the GRBF kernel**
- **Learning α in matrix α-rooting normalisation**

ACC (%)	MIT indoor	Cars	Aircraft	Birds
Initial θ	0.1	0.1	0.1	0.1
Initial α	0.5	0.5	0.5	0.5
Final θ	0.63	1.4	0.67	0.93
Final α	0.49	0.52	0.53	0.52

Research trends on learning SPD representation

- **Compactness** of second-order feature representation & Computational efficiency
- **Efficient training** of SPD structural layers by considering the underlying manifold structure
- Second-order correlation **across layers**
- **Deeply integrated** into convolutional neural networks
- **More applications** explored
    - Generic and Fine-grained image recognition
    - Image segmentation, Person reidentification and retrieval
    - Action parsing & analysis, Image super-resolution
    - More to be explored…

Conclusion

- **Discriminative Stein kernel** to address two issues in covariance representation
- **SICE representation** to incorporate structure sparsity
- **Kernel matrix representation** to move beyond linear, fixed covariance representation
- **End-to-end deep learning** of KSPD representation

1. M. Engin, L. Wang, L. Zhou, and X. Liu, DeepKSPD: Learning Kernel-matrix-based SPD Representation for Fine-grained Image Recognition, *The 15th European Conference on Computer Vision (ECCV)*, September 2018.

2. J. Zhang, L. Wang, L. Zhou, and W. Li, Learning Discriminative Stein Kernel for SPD Matrices and Its Applications, *IEEE Transactions on Neural Networks and Learning Systems (TNNLS)*, Vol. 27, Issue 5, pp. 1020-1033, May 2016.

3. L. Wang, J. Zhang, L. Zhou, C. Tang and W. Li, Beyond Covariance: Feature Representation with Nonlinear Kernel Matrices, *IEEE International Conference on Computer Vision (ICCV)*, December 2015.

4. J. Zhang, L. Wang, L. Zhou, and W. Li, Exploiting Structure Sparsity for Covariance-based Visual Representation, arXiv:1610.08619 [cs.CV].

Other related publications

- J. Zhang, L. Zhou and L. Wang, Subject-adaptive Integration of Multiple SICE Brain Networks with Different Sparsity, Pattern Recognition, 63 642-652, 2017.

- L. Zhou, L. Wang, J. Zhang, Y. Shi and Y. Gao, Revisiting Distance Metric Learning for SPD Matrix based Visual Representation, IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR), July 2017.

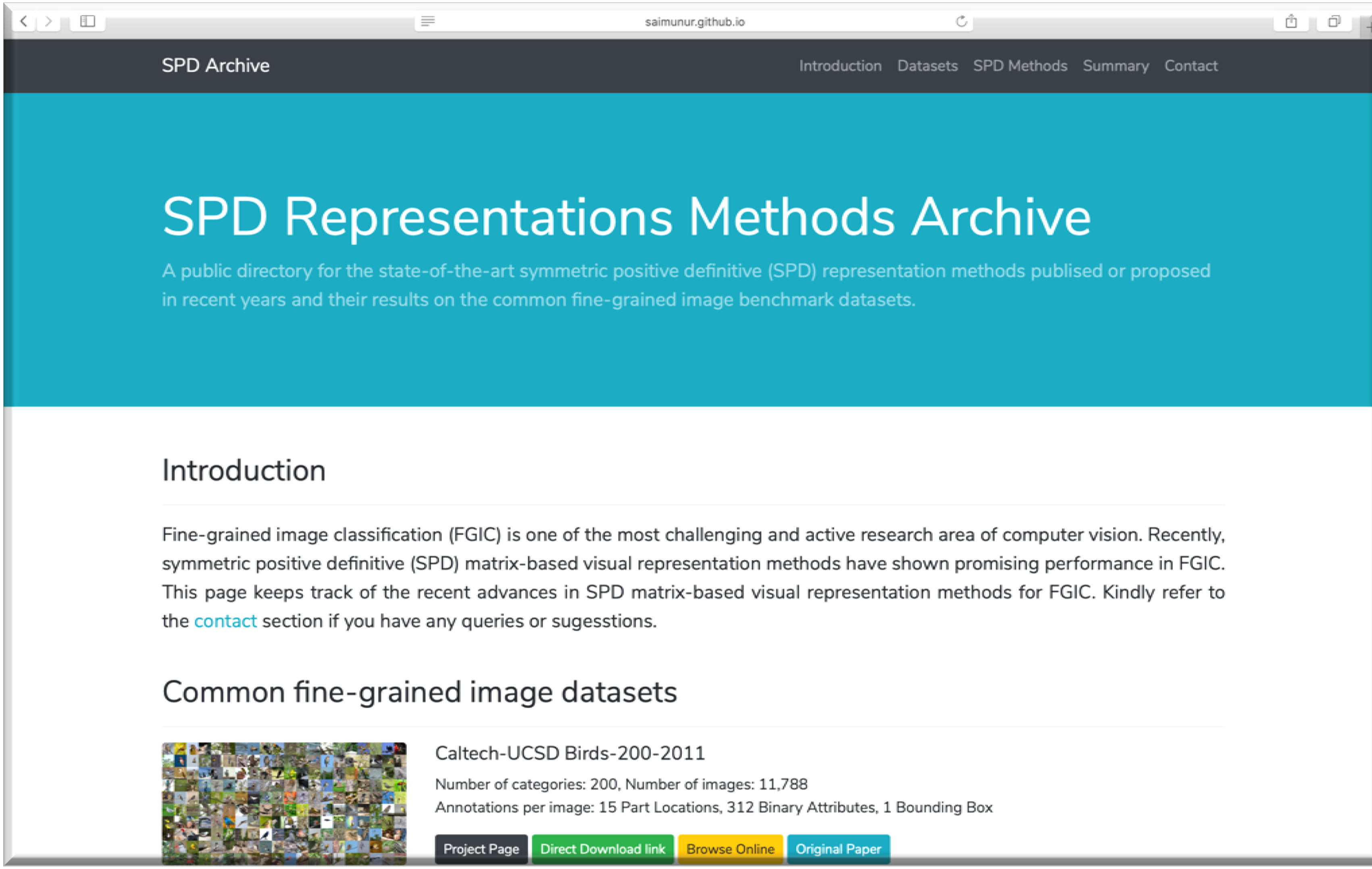
- L. Zhou, L. Wang, L. Liu, P. Ogunbona, and D. Shen, Learning Discriminative Bayesian Networks from High-dimensional Continuous Neuroimaging Data, IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI), Volume: 38 , Issue: 11 , Nov. 1 2016 .

- J. Zhang, L. Zhou, L. Wang, and W. Li, Functional Brain Network Classification With Compact Representation of SICE Matrices, IEEE Transactions on Biomedical Engineering, 62 (6), 1623-1634, 2015.

- L. Zhou, L. Wang and P. Ogunbona. Discriminative Sparse Inverse Covariance Matrix: Application in Brain Functional Network Classification, IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR), June 2014

- L. Zhou, L. Wang, L. Liu, P. Ogunbona and D. Shen. Max-margin Based Learning for Discriminative Bayesian Network from Neuroimaging Data, In the 17th International Conference on MICCAI, September 2014.

An archive website



https://saimunur.github.io/spd-archive/

Q&A

Images Courtesy of Google Image