Eigenface Recognition with Deep PCA

Presenters:
Yulai Zhao, Xiaxin Shen

Motivation

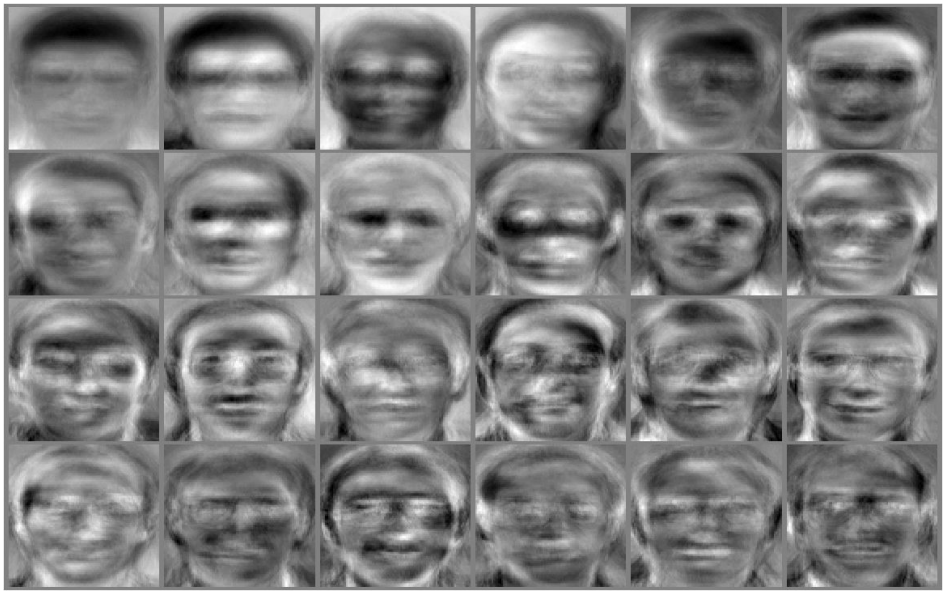
- Principal Component Analysis (PCA) is a mathematical technique used for reducing the dimensionality of high-dimensional data.
- Eigenfaces are the eigenvectors of the covariance matrix that are computed using PCA on a set of face images.
- As we know, Deep PCA is useful for abstracting features of CNN which is a better choice than MLPs in the field of computer vision.
- Use Deep PCA in finding eigenfaces

Related Work

- Eigenfaces
- PCA: classical feature extractors, useful for MLPs
- RCA: a broader component analysis that includes PCA and LSE.
- DeepPCA: built for CNNS

Eigenface

- Reduces each face image to a vector
- Then uses Principal Component Analysis (PCA) to find a set of low-dimensional features that capture the most important variations in the data
- Each eigenface represents a direction in the high-dimensional space of face images that accounts for a significant amount of variation in the data.



(a)

(b)

Figure 1: (a) Face images used as the training set. (b) The average face Ψ.

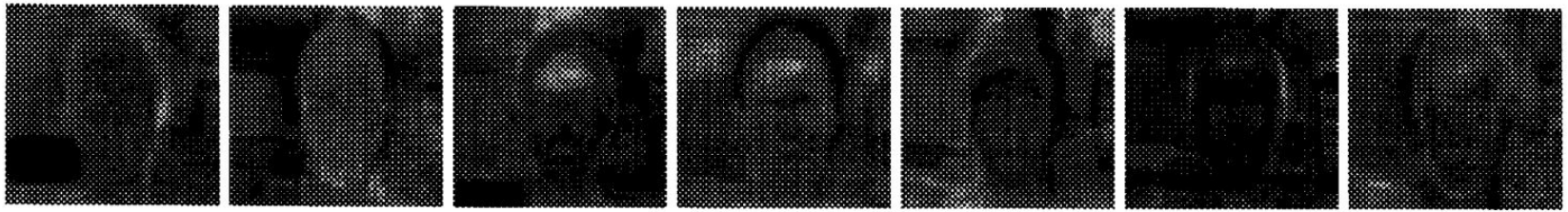


Figure 2: Seven of the eigenfaces calculated from the images of Figure 1, without the background removed.

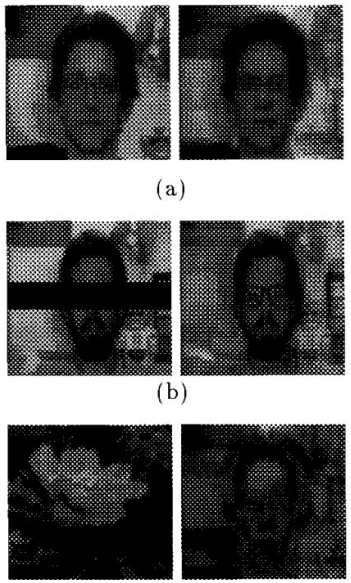
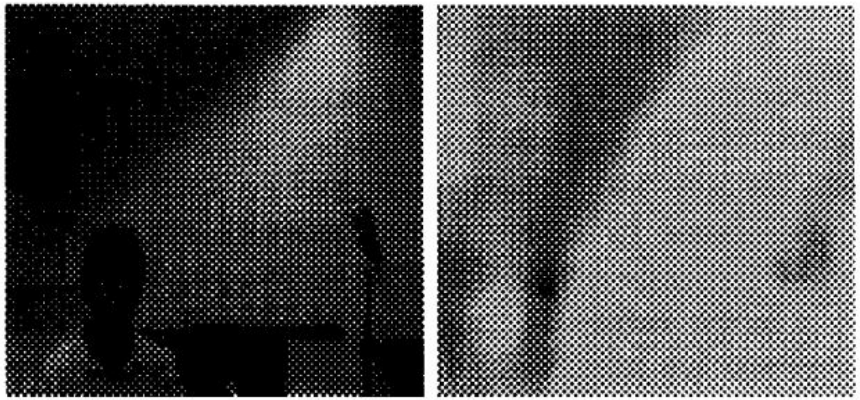


Figure 3: Three images and their projections onto the face space defined by the eigenfaces of Figure 2.



(a) (b)

Figure 4: (a) Original image. (b) Face map, where low values (dark areas) indicate the presence of a face.

Related work of Eigenface

Eigenfaces have been widely researched and developed since it was first introduced by Sirovich and Kirby in 1987. There have been numerous notable works related to eigenfaces.

- Turk and Pentland (1991): Turk and Pentland were the first to introduce eigenfaces for face recognition in 1991.
- Moghaddam and Pentland (1997): Moghaddam and Pentland introduced a modified version of eigenfaces, called Fisherfaces, that used linear discriminant analysis to improve the recognition accuracy of eigenfaces.
- Lee and Kuo (1997): Lee and Kuo introduced a variant of eigenfaces called Independent Component Analysis (ICA) that used a nonlinear transformation to extract features that were statistically independent.
- Yang et al. (2002): Yang et al. proposed a method called kernel eigenfaces, which used kernel principal component analysis to extract nonlinear features from face images.

Related work of Eigenface

- Ahonen et al. (2006): Ahonen et al. proposed a method called Local Binary Patterns (LBP) that used local texture information to represent faces.
- Huang et al. (2014): Huang et al. proposed a deep learning approach to face recognition called DeepFace, which used a convolutional neural network to extract features from face images. DeepFace was shown to outperform traditional methods like eigenfaces and LBP on several face recognition benchmarks.

To this end, a natural question is: can we both utilize CNN to extract features and retain the explainability of component analysis?

Unsupervised PCA for Dimension Reduction.

- RM is degenerated into a symmetric covariance matrix
- PCA(X) is typically formed from the principal eigenvectors

$$\mathrm{RM} = [\mathbf{X}\mathbf{X}^T]^{-1} \left(\mathbf{X}\mathbf{X}^T \, \mathbf{X}\mathbf{X}^T\right) = \mathbf{X}\mathbf{X}^T.$$

Supervised RCA for Dimension Reduction.

$$\mathcal{L}(\mathbf{F}, \mathbf{U}) = \| \mathbf{Y} - \mathbf{FUX} \|_F^2, \tag{7}$$

where the subscript F denotes the Frobenius norm and $\mathbf{F} \in \Re^{L \times m}$ and $\mathbf{U} \in \Re^{m \times M}$ with $m \leq \min\{M, L\}$. The optimal RCA solution involves a two-stage process in **min-min** optimization:

$$\min_{\mathbf{F},\mathbf{U}} \mathcal{L}(\mathbf{F}, \mathbf{U}) = \min_{\mathbf{U}} \left[\min_{\mathbf{F}} \mathcal{L}(\mathbf{F}, \mathbf{U}) \right]. \tag{8}$$

Note that its inner optimizer (for any fixed matrix $\mathbf{U}$) can be solved via the conventional LSE method (cf. Equation (2)). Thus, we have

$$\mathbf{F} = [\mathbf{UXX}^T\mathbf{U}^T]^{-1}\mathbf{UXY}^T. \tag{9}$$

Plugging Equation (9) into Equation (7), we arrive at

$$\mathcal{L}(\mathbf{F}, \mathbf{U}) = \|\mathbf{Y}\|_F^2 - tr\left[\left(\mathbf{U}[\mathbf{XX}^T]\mathbf{U}^T \right)^{-1} \left(\mathbf{UXY}^T \ \mathbf{YX}^T\mathbf{U}^T \right) \right], \tag{10}$$

whose minimum can be attained by setting $\mathbf{U}$ as follows:

$$\mathbf{U} = [\mathbf{v}_1, \ldots, \mathbf{v}_m]^T, \tag{11}$$

where $\{\mathbf{v}_1, \ldots, \mathbf{v}_m\}$ are the m principal eigenvectors of the RM, $[\mathbf{XX}^T]^{-1} \left(\mathbf{XY}^T \ \mathbf{YX}^T \right)$ (cf. Equation (4)). Thus, we have the following RCA theorem.

THEOREM 1 (RCA THEOREM: RCA(**X**,**Y**)). *The optimal RCA estimate can be expressed as* $\hat{\mathbf{Y}} = \mathbf{H}^T\mathbf{X}$, *where* $\mathbf{H}^T = \mathbf{FU}$ *and*

- *According to Equation (11),* $\mathbf{U} = [\mathbf{v}_1, \ldots, \mathbf{v}_m]^T$.
- *Thereafter, via Equation (9), we have* $\mathbf{F} = \mathbf{YX}^T\mathbf{U}^T[\mathbf{UXX}^T\mathbf{U}^T]^{-1}$.

Plugging Equations (9) and (11) into Equation (10), we have

$$LSE_{RCA} = \|\mathbf{Y}\|_F^2 - \sum_{i=1}^{m} \lambda_i . \tag{12}$$

Example 1 (LSE, RCA). Let the input and output data matrices be given as follows:

$$\mathbf{X} = \begin{bmatrix} 1 & 0.1 & 0 & 0 & 0 \\ 0 & 1.1 & 0 & 0 & 0 \end{bmatrix} \text{ and } \mathbf{Y} = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 \\ 0 & 3 & 2 & 1 & 0 \end{bmatrix}$$

(a) *LSE solution*: The optimal LSE solution (with a full rank $m = 2$) is $\mathbf{H}^T = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$, which leads to an LSE of $\text{LSE}_{LSE} = 8$.

(b) *PCA solution*: The eigenvalues and eigenvectors of $\text{RM} = \begin{bmatrix} 1 & 1 \\ 1 & 10 \end{bmatrix}$ are

$$\lambda_1 = 10.11,\ \mathbf{v}_1 = [0.1212\ 0.9926\ ]^T;\ \text{and}\ \lambda_2 = 0.89, \mathbf{v}_2 = [-0.9823\ 0.1873]^T.$$

Thereafter, via Equations (9) and (11), the optimal rank-1 projection matrix can be obtained as

$$\mathbf{H}^T = \mathbf{FU} = \begin{bmatrix} 0.1204 & 0.9859 \\ 0.3254 & 2.6652 \end{bmatrix}.$$

This ultimately leads to the following estimation error: $\text{LSE}_{RCA} = \text{LSE}_{LSE} + \lambda_2 = 8 + 0.89 = 8.89$.

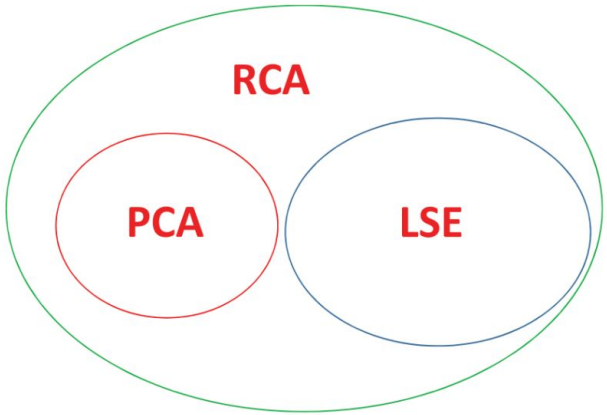


Fig. 4. Two vital special cases of RCA: (1) PCA (when $\mathbf{X} = \mathbf{Y}$) and (2) LSE (when $\mathbf{H}$ is fully ranked). Historically, Adrien-Marie Legendre and Carl Friedrich Gauss were both the recognized pioneers of the least squares method in the early 19th century. However, Gauss was also credited for systematically laying computational and theoretical foundation for LSE method. As to PCA, while the full phrase "Principal Component Analysis" was coined by Hoteling [49], its theoretical foundation PCA was first developed by Pearson [48].

Figure 4 depicts two important special cases of RCA: (1) PCA (PCA is the same as RCA when $\mathbf{X} = \mathbf{Y}$) and (2) LSE (RCA becomes LSE when $\mathbf{H}$ is fully ranked). The latter leads to the conventional LSE of $\text{LSE}_{LSE} = \|\mathbf{Y}\|_F^2 - \sum_{i=1}^{M} \lambda_i$, since all of the eigen-components are fully accounted for.

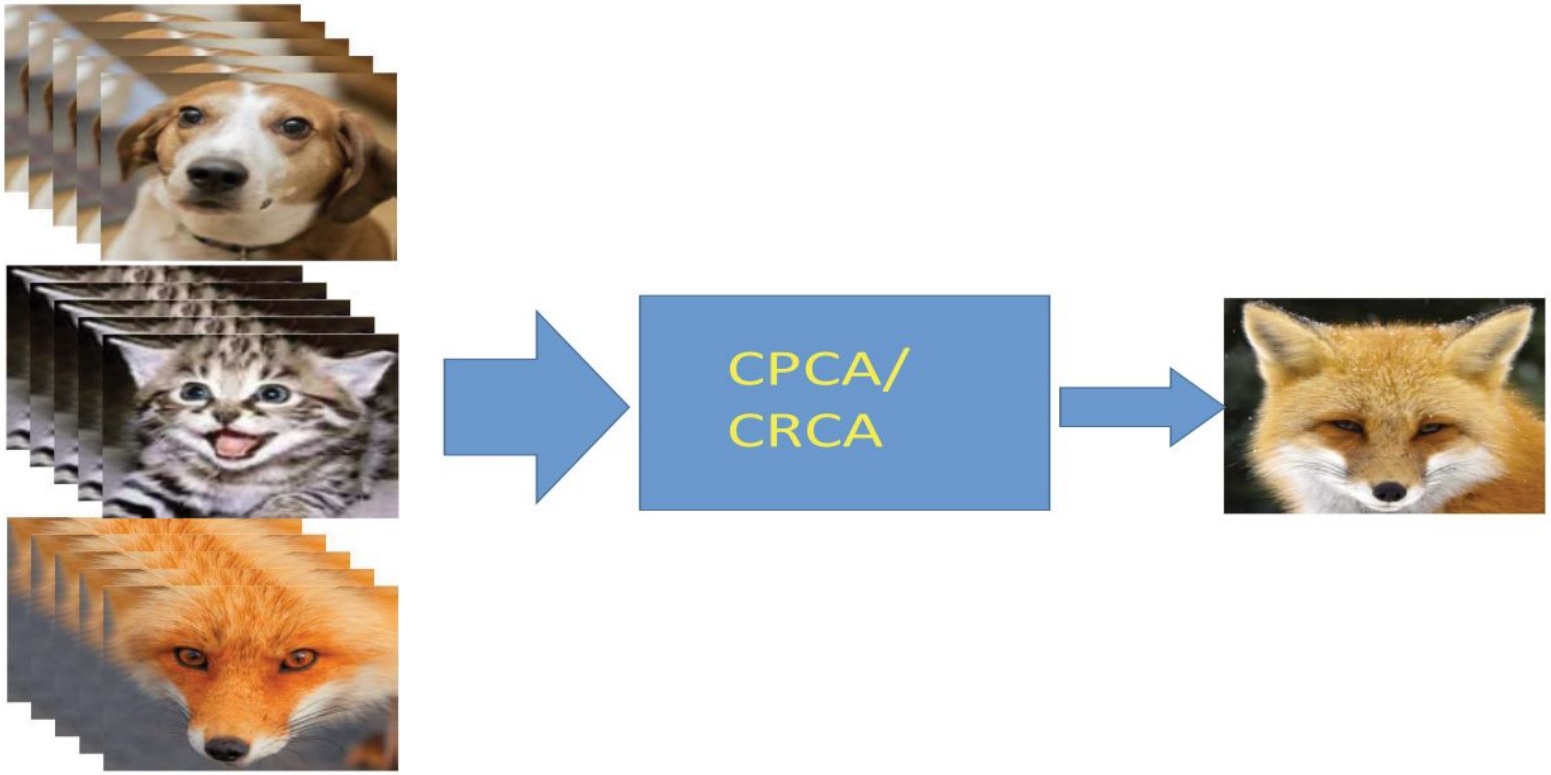
DPCA (Convolutional-PCA (CPCA))

$$\mathbf{y}_{DPCA} = \vec{\mathbf{W}}\,\mathbb{X}_{STEM}$$

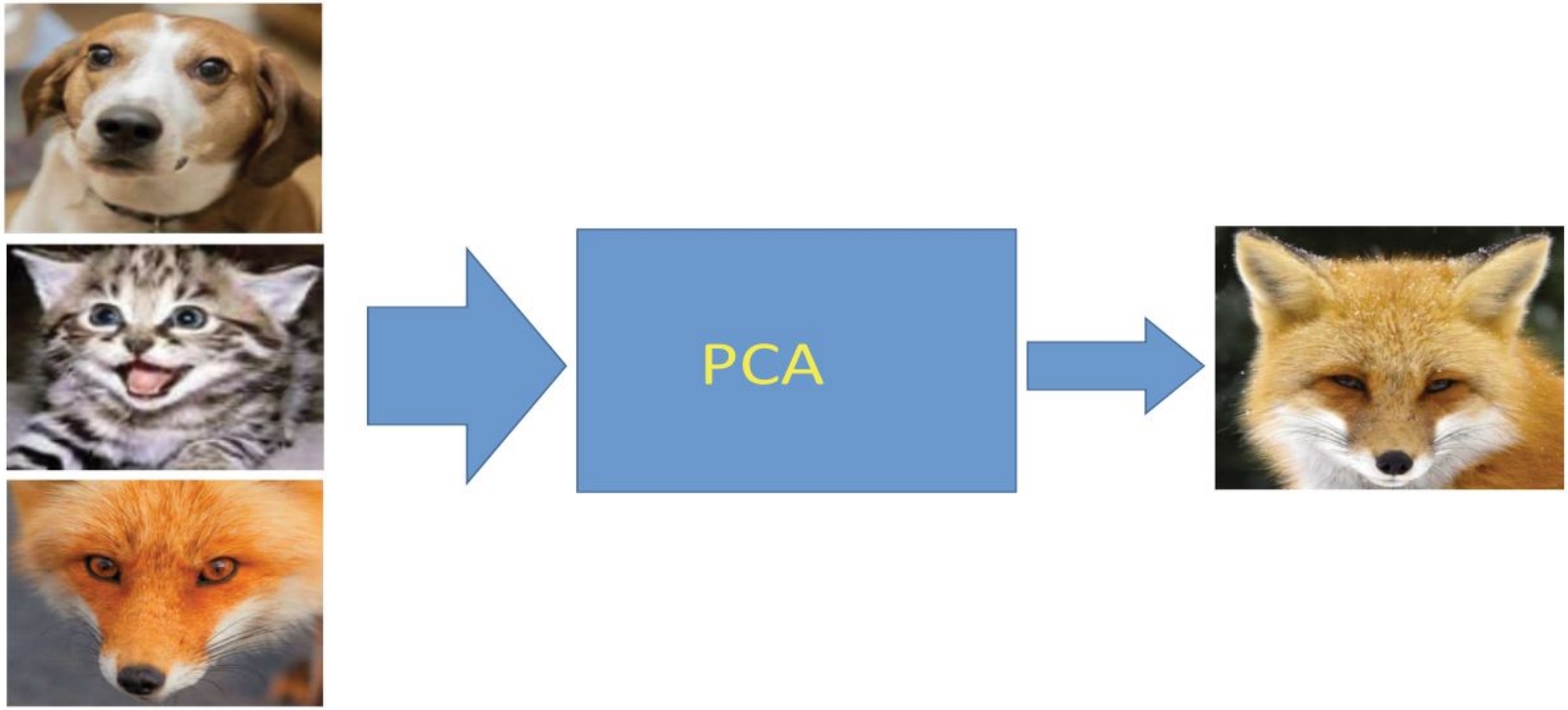
$$= \begin{bmatrix} \vec{\mathbf{w}}_{1,1} & \vec{\mathbf{w}}_{1,2} & \cdots & \cdots & \vec{\mathbf{w}}_{1,C_{in}} \\ \vec{\mathbf{w}}_{2,1} & \vec{\mathbf{w}}_{2,2} & \cdots & \cdots & \vec{\mathbf{w}}_{2,C_{in}} \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ \vec{\mathbf{w}}_{m,1} & \vec{\mathbf{w}}_{m,2} & \cdots & \cdots & \vec{\mathbf{w}}_{m,C_{in}} \end{bmatrix} \begin{bmatrix} \mathcal{X}_1(1) & \mathcal{X}_1(2) & \cdots & \mathcal{X}_1(B) \\ \mathcal{X}_2(1) & \mathcal{X}_2(2) & \cdots & \mathcal{X}_2(B) \\ \vdots & \vdots & \vdots & \\ \mathcal{X}_{C_{in}}(1) & \mathcal{X}_{C_{in}}(2) & \cdots & \mathcal{X}_{C_{in}}(B) \end{bmatrix} \quad (35)$$

The matrix $\mathbb{X}_{STEM}$, referred to as the **Space Time Expanded Matrix (STEM)** associated with 1D or 2D feature maps, has large dimensions both vertically (space-wise) and horizontally (time-wise):

$$\mathbb{X}_{STEM} \in \mathfrak{R}^{\kappa C_{in} \times D'B}$$


CPCA/
CRCA

(a)



(b)

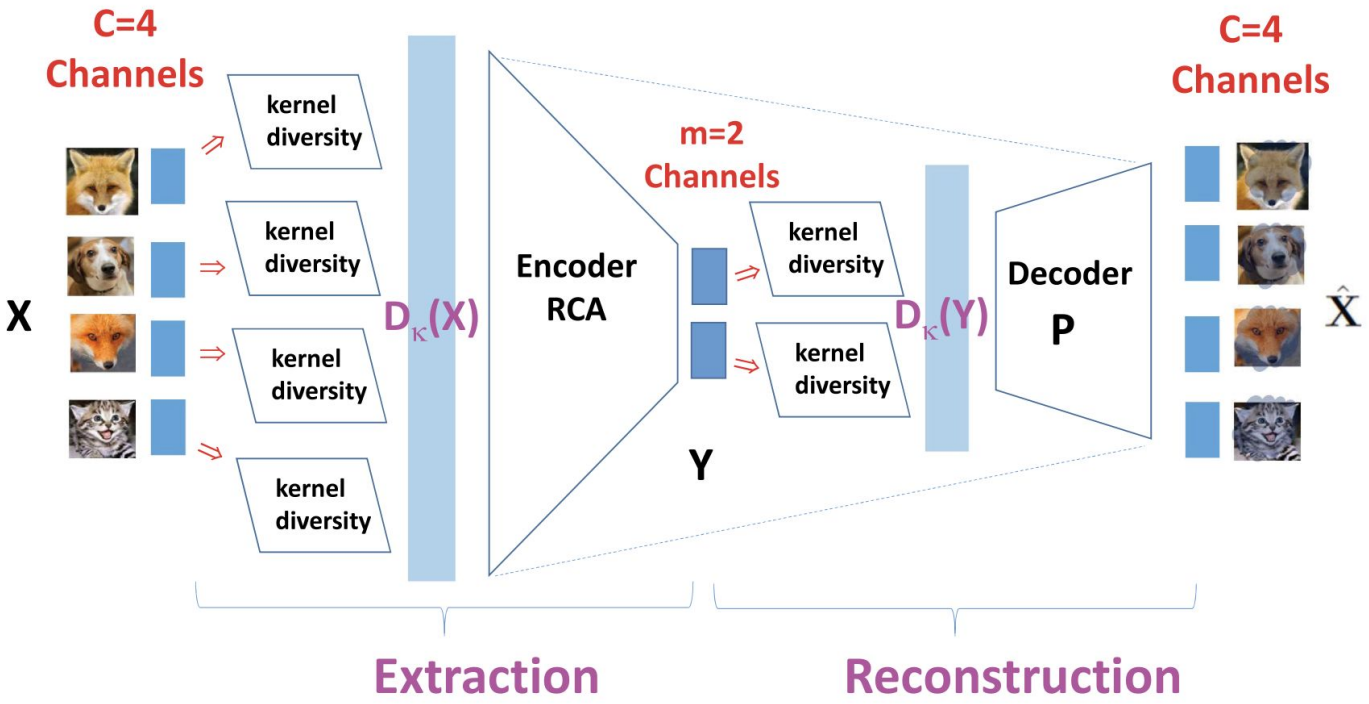


Fig. 10. For RCA-DPCA, the input diversity ($[\mathbf{D}_k\mathfrak{X}]\ ||_F$) is used as input data array. However, the reconstructed space and the LSE are represented/measured by the non-diversified space spanned by $\mathfrak{X}$.

DPCA is based on the following two-stage optimization formulation

- Encoder or extraction phase: Extraction of DPCA.
- Decoder or reconstruction phase: Reconstruction from DPCA.

DPCA

3.2.1 Encoder Phase: Formulation for Extraction of DPCA. The extraction phase of DPCA is pictorially illustrated by Figure 10. Let the optimal estimate be denoted as $\mathfrak{Y} = \mathrm{W}[\mathbf{D}_k\mathfrak{X}]$, and we seek an optimal matrix $\mathbf{W}$, with $\mathbf{W} \in \Re^{C_{in} \times \kappa C_{in}}$ and $\mathrm{rank}(\mathbf{W}) = m \leq C_{in}$, such that

$$\min_{\mathbf{W}} || \mathfrak{X} - \mathbf{W} [\mathbf{D}_k\mathfrak{X}] ||_F. \tag{36}$$

In this case, the goal is to make $\mathfrak{Y}$ approach $\mathfrak{X}$ as closely as possible. However, $\mathfrak{Y}$ is extracted from $\mathbf{D}_k\mathfrak{X}$ instead of $\mathfrak{X}$. Such an input-output asymmetry necessitates the adoption of the RCA algorithm instead of the conventional PCA. In fact, when ambiguity is a concern, it would be better to explicitly refer to DPCA of such kind as RCA-DPCA.

3.2.2 Decoder Phase: Formulation for Reconstruction from DPCA. In the reconstruction phase (cf. Figure 10), it is vital to harness the *kernel diversity* associated with the extracted components: $\mathfrak{Y}$. (Namely, the subspace used for reconstruction should be $\mathbf{D}_k\mathfrak{Y}$ instead of $\mathfrak{Y}$.) Furthermore, there exist two plausible formulations for reconstruction, each with its own justificator:

- *Reconstruction of non-diversified space* $\mathfrak{X}$: In this case, we treat DPCA as an end product. This corresponds to the scenario when the augmenting hidden layer (cf. Figure 6) happens to be the final layer and no more convolutions are anticipated. This calls for a typical LSE formulation to derive the best matrix $\mathbf{P} \in \mathfrak{R}^{C_{in} \times \kappa m}$ such that

$$\min_{\mathbf{P}} || \mathfrak{X} - \mathbf{P} [\mathbf{D}_k\mathfrak{Y}] ||_F. \tag{38}$$

  This type of reconstruction is pictorially illustrated by Figure 10.

Alternatives

Alternatively, there may be situations that we would want the estimate, $\mathfrak{Y} = \mathbf{W}[\mathbf{D}_k\mathfrak{X}]$, where $\mathbf{W} \in \mathfrak{R}^{\kappa C_{in} \times \kappa C_{in}}$ and $\text{rank}(\mathbf{W}) = m \leq C_{in}$, to best approach $\mathbf{D}_k\mathfrak{X}$ instead of $\mathfrak{X}$. In this case, the optimization formulation is modified to

$$\min_{\mathbf{W}} || \mathbf{D}_k\mathfrak{X} - \mathrm{W}[\mathbf{D}_k\mathfrak{X}] ||_F \tag{37}$$

Alternatives

- *Reconstruction of diversified space* $\mathbf{D}_k\mathfrak{X}$: In this case, the extracted DPCA is no longer treated as an end product. Therefore, some further convolutions and diversification will be expected. This corresponds to the scenario that the augmented hidden layer (cf. Figure 6) is positioned in the middle and supposed to continue on with further convolution-type processing. In this case, we will need to consider a different LSE formulation, seeking $\mathbf{P} \in \Re^{\kappa C_{in} \times \kappa m}$ for

$$\min_{\mathbf{P}} || \mathbf{D}_k\mathfrak{X} - \mathbf{P}[\mathbf{D}_k\mathfrak{Y}] ||_F. \tag{39}$$

Why Use DPCA on FBI images

- The importance of explainability
- Saving storage space

Future Work

- Neural Architecture Search (NAS)
- X-learning learning paradigm
  - X-learning is proposed as a joint parameter and structural learning paradigm
  - Structural pruning

X-Learning	Low-Power	High-Performance
Classification	XMobileNetV2 XMNasNet	XResNet50 XResNet164
Regression	XSR-ResNet	LapSRN XSR-ResNet