CausalX: Causal eXplanations and Block Multilinear Factor Analysis Appendix

M. Alex O. Vasilescu[1,2]
maov@cs.ucla.edu

Eric Kim[2,1]
ekim@cs.ucla.edu

Xiao S. Zeng[2]
stevennz@ucla.edu

[1]Tensor Vision Technologies, Los Angeles, California
[2]Department of Computer Science,University of California, Los Angeles

Appendix

A. Image as a Vector, Matrix, or Tensor?

Raw observational data usually comprise sets of random variables. Historically, multivariate data analysis has been performed on observations organized both as a "data matrix" [1] and as a vector [21], [24].

In statistical analysis, organizing a multivariate observation as a data matrix gave way to organizing a multivariate observation as a vector of measurement variables. In recent years, once again, several authors have advocated leaving the data elements associated with a single observation organized in the manner that the acquisition device provides them; in particular, organizing a CCD image as a matrix of pixel variables, rather than "vectorizing" it. In this context, linear and multilinear tensor models for vision, graphics, and learning fall under four different categories:

1) Multilinear, or rank-$(R_1, R_2, \ldots, R_M)$, decomposition and different generalizations for higher-order data tensors that contain vectorized measurement data, as in our work [25], [27] and the work of other authors [33], [29], [7].
2) Like Category 1 above, but with rank-R decomposition.
3) Multilinear, or rank-$(R_1, R_2, \ldots, R_M)$, decomposition of higher-order data tensors where each observation is a matrix or higher-order tensor; e.g., an image is treated as a matrix and a unimodal image ensemble is a third-order tensor [36], [22], [34], [31], [32], [9].[1]
4) Like Category 3 above, but with rank-R decomposition [19], [3], [30], [18].

CNN tensor factorization methods employ a image as matrix/tensor approach to reducing the number of parameters [13], [15], [11], [5], [4], [16], [17].

Several of the above papers have asserted:

- *"An image (video) is intrinsically a matrix (tensor)."*
- *"Images, for example, are naturally represented as third order tensors, where the modes correspond to height, width, and channels,"*

Technically, a matrix corresponds to a transformation from one vector space to another (a multilinear transform is a mapping from a set of vector spaces to a vector space), so a more appropriate statement would be that an image (video) is a 2D (3D) array of numbers. The photoreceptors in biological eyes are by no means organized as regular matrices or tensors. This statement merely presupposes a CCD imaging sensor with a rectangular array of receptors. Yet one can readily treat both cases in a uniform manner by vectorizing the collection of receptor responses, thus treating the *entire* image as a point in a high-dimensional image space.

- *" An inherent problem of the image-as-a-vector representation is that the spatial redundancy within each image matrix is not fully utilized, and some local spatial relationships are lost."*

On the contrary, the subdivision of the image into rows and columns, suffers from the very problem incorrectly attributed to the conventional, vectorized-image representation. When an image is treated as a matrix, from a statistical perspective it becomes an ensemble of row/column measurement variables which results in computing a subset of all possible variable combinations, Figure 1. Most arguments in favor of the latter approach that are found in the literature are provably false.

There are two major drawbacks with the treatment of image-as-a-matrix. Performing dimensionality reduction in the row or column space can result in throwing out relevant object identity information. Second, one of the major limitation of treating an image-as-a matrix instead of computing one object representation regardless of other extrinsic causal factors, there are as many representations as there are images for an individual in the database. The number of images per person in a database may grow combinatorially with the number of causal factors that could form an image. In order to overcome this type of growth additional mathematical machinery is needed to minimize intra-class representation scatter [23].

- *"We overcome the curse of dimensionality by treating images as matrices" (i.e., as a set of separate column measurements rather than a single measurement vector).*
- *"First, in real applications such as face recognition, a very limited number of sample images are typically available for each subject, resulting in the well-known small-sample-size problem. In such contexts, an image-as-matrix representation is more appropriate, since the*

[1] Some authors [36], [31] claim that they perform a generalized rank-R approximation of tensors; however, according to the standard definition of the rank-R and rank-$(R_1, R_2, \ldots, R_M)$ decompositions, they actually perform a multilinear decomposition of a third-order tensor with dimensionality reduction in only two of the modes—the image row/column modes. This multilinear decomposition is also known as a Tucker2 decomposition or a $(M-1)$-mode SVD.

smaller number of data entries along each data dimension facilitates subspace learning from little training data."

Treating an image as matrix does not resolve or directly deal with the curse-of-dimensionality and/or small-sample-size problems. Consider the extreme case where the sample size is a single image. Clearly, one image provides no statistically significant information about the facial image space, and treating an image as a matrix—i.e. the image as a set of observations, where each row/column is a measurement—will not provide any further information about the facial image space. It simply computes a set of statistics associated with the rows/columns of that single image. In the scenario where an image is treated as a matrix, as in the case of the 2DPCA paper[35], the columns are being treated as if they are independent and interchangeable parts for which the average column statistics are being computed. Hence, the resulting generative model might result in an image with the same set of columns repeated, *i.e.*, a facial image with multiple eyes or noses.

- *"Treating an image as a matrix, results in a model that uses less storage."*

The storage requirements of both PCA and 2DPCA [35] grow linearly with the number of images, while the amount of storage needed to store the basis vectors is constant. This statement applies both to 2DPCA which performs a 1-mode SVD of a third-order tensor, and to 2DSVD [36], which performs a 2-mode SVD of a third-order tensor; in both instances, the tensor is an ensemble of 2D images. The upper bound on the PCA basis storage requirements (corresponding to no dimensionality reduction) is the number of pixels squared, while that of the 2DPCA is the number of rows squared plus the number of columns squared. While the basis vector storage requirements for 2DPCA is less than that of PCA, the dominant factor is the amount of storage needed to represent the images in the database. By contrast, the storage requirements of MPCA and MICA grow linearly with the number of people, which is usually only a fraction of the number of images which grow combinatorially with the number of causal factors of data formation.

- *"Computing PCA on a dataset that treats an image as a matrix is computationally more efficient."*

The PCA of a data matrix of vectorized $N_1 \times N_2$ images computes the SVD of the pixel covariance matrix whose dimensionality is $(N_1 N_2)^2$. When treating images as "matrices" and an image ensemble as a third-order tensor, one obtains a column-pixel covariance matrix and a row-pixel covariance matrix, of dimensionality N_1^2 and N_2^2, respectively. Computing the SVDs of two smaller covariance matrices for 2DSVD is less expensive than performing an SVD on a large covariance matrix for PCA, but one should bear in mind that this is an offline computational task. Our MPCA/MICA algorithms treat images as vectors, but they compute SVDs not of the pixel covariance matrix, but of the covariance matrices associated with each causal factor. Although this can be more expensive than PCA or 2DSVD, depending on the size of the causal factor matrices, it is an offline computational cost that is offset by a less expensive online recognition computation. Several papers in the literature advocate the computation of the MPCA by treating images as matrices (e.g., [31], [32]), which additionally necessitates computing SVDs of the row-pixel covariance and column-pixel covariance matrices.

Mathematical analysis: We examine the mathematical relationship between image-as-a-vector and image-as-a-matrix representations in more detail. A data matrix $\mathbf{D} \in \mathbb{R}^{I_{\text{x}} \times I_{\text{I}}}$ whose columns are vectorized images (Figure 1(a)) can be decomposed using standard matrix SVD.

PCA, which employs the matrix SVD, applied to an ensemble of vectorized images encodes the pixel covariances or second-order statistics of the ensemble. In other words, treating an $I_{\text{xc}} \times I_{\text{xr}}$ image as a vector leads to the computation for each pixel of *all* its pairwise covariances with *every other* pixel in the image (there are $I_{\text{x}}^2 = (I_{\text{xc}} I_{\text{xr}})^2$ such covariances for each pixel) (Figure 1(a)). By contrast, when one regards an image as a matrix, it becomes a collection of row (column) observations and PCA explicitly computes how pixels co-vary with other pixels within the same row (column).

The SVD is expressed in terms of the mode-m product and matrix multiplication as follows:

$$\begin{aligned} \mathbf{D} &= \mathbf{S} \times_{\text{x}} \mathbf{U}_{\text{x}} \times_{\text{I}} \mathbf{U}_{\text{I}} && \text{Tensor Notation} \quad (1) \\ \mathbf{D}_{[\text{x}]} = \mathbf{D}_{[\text{I}]}^{\text{T}} &= \mathbf{U}_{\text{x}} \underbrace{\mathbf{S}\mathbf{U}_{\text{I}}^{\text{T}}}_{\mathbf{R}^{\text{T}}}, && \text{Matrix Notation} \quad (2) \end{aligned}$$

where $\mathbf{U}_{\text{x}}$ contains the PCA basis vectors. These basis vectors are computed from the pixel covariance matrix, which contains all possible pairwise pixel covariances (Figure 1(a)). Matrix $\mathbf{R}$ is the response/coefficient matrix, where the i column in $\mathbf{R}^{\text{T}}$ is the coefficient representation relative to the PCA basis of $\mathbf{d}_i$, which is the i observation/column in $\mathbf{D}$.

When an $I_{\text{xc}} \times I_{\text{xr}}$ image is organized as a data matrix $\mathbf{D}_i$ and an ensemble of such images is organized into a third order data tensor $\mathcal{D} \in \mathbb{R}^{I_{\text{xc}} \times I_{\text{xr}} \times I_{\text{I}}}$ (Figure 1(d)), the data tensor can be decomposed using the M-mode SVD as follows:

$$\begin{aligned} \mathcal{D} &= \mathcal{Z} \times_{\text{xc}} \mathbf{U}_{\text{xc}} \times_{\text{xr}} \mathbf{U}_{\text{xr}} \times_{\text{I}} \mathbf{U}_{\text{I}} && (3) \\ &= \mathcal{R} \times_{\text{xc}} \mathbf{U}_{\text{xc}} \times_{\text{xr}} \mathbf{U}_{\text{xr}}, && (4) \end{aligned}$$

where $\mathbf{U}_{\text{xc}}$ is computed by matrixizing the data tensor along the row (Figure 1(b)) and computing the basis vectors associated with the covariance matrix between pixels within image columns, where each column is considered a single measurement. Matrix $\mathbf{U}_{\text{xr}}$ contains the basis vectors associated with the covariance matrix between pixels within image rows, where each row is considered a single measurement (Figure 1(c)). The product $\mathcal{R} = \mathcal{Z} \times_{\text{I}} \mathbf{U}_{\text{I}}$ contains the response for each image when projected onto $\mathbf{U}_{\text{xc}}$ and $\mathbf{U}_{\text{xr}}$. $\mathbf{R}_i$ is the i slice of tensor $\mathcal{R}$ and it contains the response or coefficients associated with image $\mathbf{D}_i$.

Matrixizing the data tensor along the image mode results in a data matrix $\mathbf{D}_{[\text{I}]}$ whose transpose is the same matrix used by PCA in Equation (2), thus giving us an opportunity to compare terms: From Equations (3) and (2), we have

$$\begin{aligned} \mathbf{D}_{[\text{I}]}^{\text{T}} &= \mathbf{U}_{\text{x}} \mathbf{S} \mathbf{U}_{\text{I}}^{\text{T}} && (5) \\ &= \underbrace{(\mathbf{U}_{\text{xc}} \otimes \mathbf{U}_{\text{xr}}) \mathbf{Z}_{[\text{I}]}^{\text{T}}}_{\substack{\text{Unnormalized} \\ \text{PCA Basis Matrix}}} \quad \underbrace{\mathbf{U}_{\text{I}}^{\text{T}}}_{\substack{\text{Normalized} \\ \text{PCA Coefficient Matrix}}} && (6) \end{aligned}$$

Essentially, M-mode SVD has decomposed the unnormalized PCA basis $(\mathbf{U}_{\text{x}}\mathbf{S})$ into two basis matrices, $\mathbf{U}_{\text{xc}}$ and $\mathbf{U}_{\text{xr}}$, whose columns span column/row pixel covariances. By contrast, when images are vectorized and organized into a multifactor tensor (akin to the TensorFaces approach), M-mode SVD further decomposes the PCA coefficient matrix.

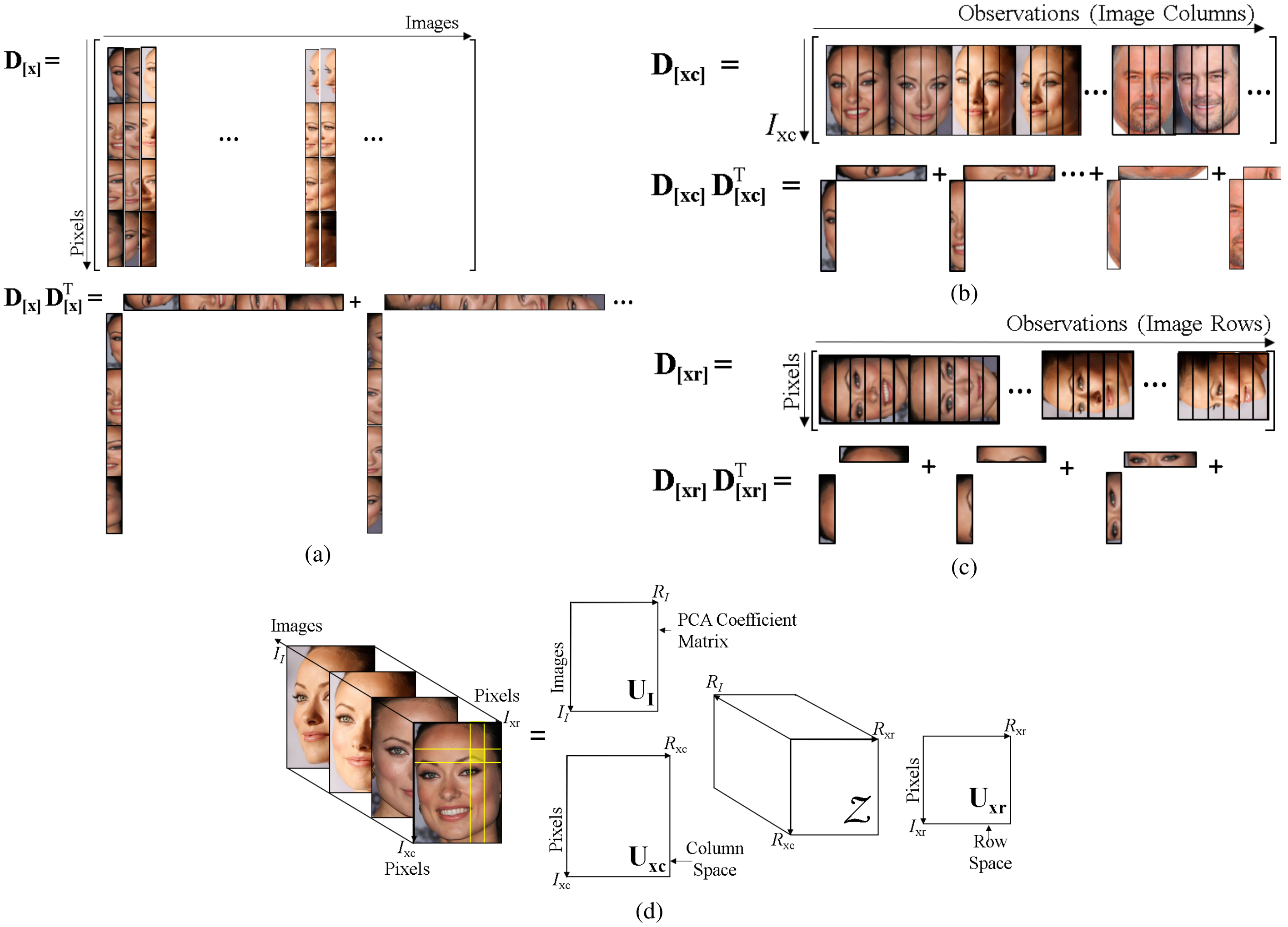

Fig. 1: (a) When the pixels associated with an image are organized into a 1-way array, PCA computes all possible pairwise pixel covariances. (b) The data tensor is matrixized in the first mode and pixel covariances are computed between pixels within a column. (c) The data tensor is matrixized in the second mode and pixel covariances are computed between pixels within a row. (d) Matrixizing a data tensor and covariance computation. (c) Image-as-a-matrix decomposition tensor decomposition. Images are projected into column space, $\mathbf{U}_{xc}$ and row space, $\mathbf{U}_{xc}$. Diagonal covariances are not computed. In the image-as-matrix scenario, a multilinear (Tucker, MICA *e*tc.) tensor factorization computes one (1) representation per image, and thus, multiple representations per person, $\mathcal{R} = \mathcal{Z} \times_{\mathrm{I}} \mathbf{U}_{\mathrm{I}}$, where $\mathcal{R}$ is the collection of image representations. Thus, in the image-as-matrix case, a Tucker or MICA tensor factorization does not result in a discriminant object representation, and additional mathematical machinery is needed to perform object recognition. Representations of the causal factors of data formation are not computed.

Tensor $\mathcal{Z}$ contains normalizing parameters associated with $\mathbf{U}_{\mathrm{I}}$, $\mathbf{U}_{\mathrm{xr}}$ and $\mathbf{U}_{\mathrm{xc}}$. It may be tempting to think of $\mathbf{Z}_{[\mathrm{I}]}^{\mathrm{T}}$ as a matrix that spans cross-diagonal pixel covariance from most important to least important; however, cross-diagonal pixel covariance is not computed, and the M-mode SVD can only model the data with respect to $\mathbf{U}_{\mathrm{xc}}$ and $\mathbf{U}_{\mathrm{xr}}$. Thus, dimensionality reduction cannot discard the least important redundancies between cross-diagonal pixels. Note that there is an exact relationship *only* if no dimensionality reduction is performed.

The same counterargument holds true for organizing a sensory input data into a higher-order tensor.

Summary: The benefits of viewing an image as a data tensor (or, as it is sometimes called, a “tensor object' / 'ND-object”[2]) have been very much overstated in well-known papers in the literature.We have argued that, instead, it is preferable to treat images as vectors and an ensemble of vectorized images as a M-way array, a “data-tensor”.

B. *Compositional Hierarchical Block TensorFaces*

Training Data: In our experiments, we employed gray-level facial training images rendered from 3D scans of 100 subjects. The scans were recorded using a CyberwareTM 3030PS laser scanner and are part of the 3D morphable faces database

[2] There exists a recent trend to refer to 3-way or N-way array as a $3D$ or ND object has overloaded the meaning of the word dimension. For example, a vectorized image is a point in high dimensional space with dimensionality $I_N \times 1$.

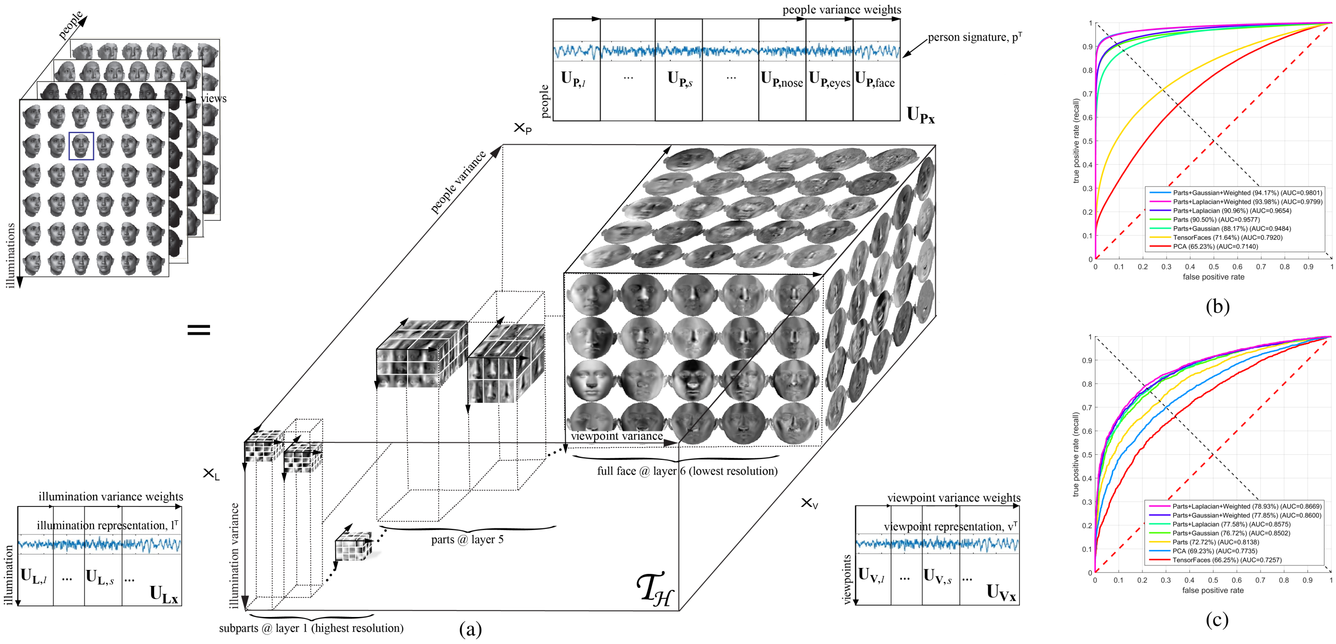
people
views
illuminations
=
people variance weights
person signature, p^T
people
$U_{P,l}$
...
$U_{P,s}$
$U_{P,nose}$
$U_{P,eyes}$
$U_{P,face}$
U_{Px}
$\times_P$
people variance
viewpoint variance
full face @ layer 6 (lowest resolution)
parts @ layer 5
subparts @ layer 1 (highest resolution)
illumination variance
$\times_L$
$\times_V$
$\mathcal{T}_{\mathcal{H}}$
illumination variance weights
illumination representation, l^T
illumination
$U_{L,l}$
$U_{L,s}$
U_{Lx}
viewpoint variance weights
viewpoint representation, v^T
viewpoints
$U_{V,l}$
$U_{V,s}$
U_{Vx}
(a)
true positive rate (recall)
false positive rate
Parts+Gaussian+Weighted (94.17%) (AUC=0.9801)
Parts+Laplacian+Weighted (93.98%) (AUC=0.9799)
Parts+Laplacian (90.96%) (AUC=0.9654)
Parts (90.50%) (AUC=0.9577)
Parts+Gaussian (88.17%) (AUC=0.9484)
TensorFaces (71.64%) (AUC=0.7920)
PCA (65.23%) (AUC=0.7140)
(b)
true positive rate (recall)
false positive rate
Parts+Laplacian+Weighted (78.93%) (AUC=0.8669)
Parts+Gaussian+Weighted (77.85%) (AUC=0.8600)
Parts+Laplacian (77.58%) (AUC=0.8575)
Parts+Gaussian (76.72%) (AUC=0.8502)
Parts (72.72%) (AUC=0.8138)
PCA (69.23%) (AUC=0.7735)
TensorFaces (66.25%) (AUC=0.7257)
(c)

Fig. 2: (a) compositional hierarchical Block TensorFaces learns a hierarchy of features, and reesents each person as a part-based compositional representation. Figure depicts the training data factorization, $\mathcal{D} = \mathcal{T}_{\mathcal{H}} \times_{\text{L}} \mathbf{U}_{\text{L}} \times_{\text{V}} \mathbf{U}_{\text{V}} \times_{\text{P}} \mathbf{U}_{\text{P}}$, where an observation is represented as $\mathbf{d}(\mathbf{p}, \mathbf{v}, \mathbf{l}) = \mathcal{T}_{\mathcal{H}} \times_{\text{L}} \mathbf{l}^{\text{T}} \times_{\text{V}} \mathbf{v}^{\text{T}} \times_{\text{P}} \mathbf{p}^{\text{T}}$ and $\mathcal{T}_{\mathcal{H}}$ spans the hierarchical causal factor variance. (b) ROC curves for the University of Freiburg 3D Morphable Faces dataset. (c) ROC curves for the LFW dataset. The average accuracies are listed next to each method, along with the area under the curve (AUC). Parts refers to using compositional hierarchical Block TensorFaces models to separately analyze facial parts. Gaussian, Laplacian refers to using compositional hierarchical Block TensorFaces on a Gaussian/Laplacian data pyramid.

Test Dataset	**PCA**	**TensorFaces**	**compositional hierarchical Block TensorFaces**				
			Pixels	Gaussian Pyramid	Weighted Gaussian Pyramid	Laplacian Pyramid	Weighted Laplacian Pyramid
Freiburg	65.23%	71.64%	90.50%	88.17%	94.17%	90.96%	93.98%
LFW	69.23% ±1.51	66.25% ±1.60	72.72% ±2.14	76.72% ±1.65	77.85% ±1.83	77.58% ±1.45	78.93% ±1.77

TABLE I: Empirical results reported for LFW : PCA, TensorFaces and compositional hierarchical Block TensorFaces. Pixels denotes independent facial part analysis Gaussian/Laplacian use a multi resolution pyramid to analyze facial features at different scales. Weighted denotes a weighted composite signature.
Freiburg Experiment:
Train on Freiburg: 6 views (±60°,±30°,±5°); 6 illuminations (±60°,±30°,±5°), 45 people
Test on Freiburg: 9 views (±50°, ±40°, ±20°, ±10°, 0°), 9 illums (±50°, ±40°, ±20°, ±10°, 0°), 45 different people
LFW Experiment: Models were trained on approximately half of one percent ($0.5\% < 1\%$) of the 4.4M images used to train DeepFace.
Train on Freiburg:
15 views (±60°,±50°, ±40°,±30°, ±20°, ±10°,±5°, 0°), 15 illuminations (±60°,±50°, ±40°,±30°, ±20°, ±10°,±5°, 0°), 100 people
Test on LFW: We report the mean accuracy and standard deviation across standard literature partitions [8], following the Unrestricted, labeled outside data supervised protocol.

created at the University of Freiburg [2]. Each subject was combinatoriall y imaged in Maya from 15 different viewpoints ($\theta = -60°$ to $+60°$ in $10°$ steps on the horizontal plane, $\phi = 0°$) with 15 different illuminations ($\theta = -35°$ to $+35°$ in $5°$ increments on a plane inclined at $\phi = 45°$).

Data Preprocessing: Facial images were warped to an average face template by a piecewise affine transformation given a set of facial landmarks obtained by employing Dlib software [12], [10], [20], [14], [6]. Illumination was normalized with an adaptive contrast histogram equalization algorithm, but rather than performing contrast correction on the entire image, subtiles of the image were contrast normalized, and tiling artifacts were eliminated through interpolation. Histogram clipping was employed to avoid over-saturated regions.

Experiments: Each image, $\mathbf{d} \in \mathbb{R}^{I_0 \times 1}$, was convolved with five filters banks $\{\mathbf{H}_s \| s = 1...S\}$. The filtered images, $\mathbf{d} \times_0 \mathbf{H}_s$, resulted in five facial part hierarchies composed of (i) independent pixel parts (ii) parts segmented from different layers of a Gaussian pyramid that were equally or (iii) unequally weighed, (iv) parts were segmented from a Laplacian pyramid that were equally or (v) unequally weighed. We ran five experiments with five facial part hierarchies from which a person representation was computed, Fig. 2. The composite person signature was computed for every test image by employing the multilinear projection algorithm[26], [28], and signatures were compared with a nearest neighbor classifier.

To validate the effectiveness of our system on real-world images, we report results on "LFW" dataset (LFW) [8]. This dataset contains 13,233 facial images of 5,749 people. The photos are unconstrained (*i.e.*, "in the wild"), and include variation due to pose, illumination, expression, and occlusion. The dataset consists of 10 train/test splits of the data. We report the mean accuracy and standard deviation across all splits in Table I. Fig. 2(b-c) depicts the experimental ROC curves. We follow the supervised "Unrestricted, labeled outside data" paradigm.

Results: While we cannot celebrate closing the gap on human performance, our results are promising. DeepFace, a CNN model, improved the prior art verification rates on LFW from 70% to 97.35%, by training on $4.4M$ images of 200×200 pixels from $4,030$ people, the same order of magnitude as the number of people in the LFW database.

We trained on less than one percent (1%) of the 4.4M total images used to train DeepFace. Images were rendered from 3D scans of 100 subjects with an the intraocular distance of approximately 20 pixels and with a facial region captured by $10,414$ pixels (image size $\approx 100 \times 100$ pixels). We have currently achieved verification rates just shy of 80% on LFW. When data is limited, CNN models do not convergence or generalize.

Summary: This paper contributes to the tensor algebraic paradigm and models cause-and-effect as a hierarchical block tensor interaction between intrinsic and extrinsic hierarchical causal factors of data formation.

A data tensor expressed as a function of a hierarchical data tensor is a unified tensor model of wholes and parts from which a new compositional hierarchical block tensor factorization was derived. The resulting causal factor representations are interpretable, hierarchical, and statistically invariant to all other causal factors. Our approach was demonstrated in the context of facial images by training on a very small set of synthetic images. While we have not closed the gap on human performance, we report encouraging face verification results on two test data sets–the Freiburg, and the Labeled Faces in the Wild datasets. CNN verification rates improved the 70% prior art to 97.35% when they employed 4.4M images from $4,030$ people, the same order of magnitude as the number of people in the LFW database. We have currently achieved verification rates just shy of 80% on LFW by employing synthetic images from 100 people for a total of less than one percent (1%) of the total images employed by DeepFace. By comparison, when data is limited, CNN models do not convergence, or generalize.

References

[1] H. Andrews and C. Patterson. Digital images as a teaching aid for singular value decomposition. Amer. Math. Month., 82:1–73, 1974.

[2] V. Blanz and T. A. Vetter. Morphable model for the synthesis of 3D faces. In Proc. ACM SIGGRAPH 99 Conf., pages 187–194, 1999.

[3] R. Furukawa, H. Kawasaki, K. Ikeuchi, and M. Sakauchi. Appearance based object modelling using texture database: Acquisition, compression and rendering. In Proc. 13th Eurographics Workshop on Rendering, pages 257–266, 2002.

[4] L. Grasedyck. Hierarchical singular value decomposition of tensors. SIAM J. on Matrix Analysis and Applications, 31(4):2019–54, 2010.

[5] W. Hackbusch and S. Kühn. A new scheme for the tensor representation. Journal of Fourier Analysis and Applications, 15(5):706–722, 2009.

[6] A. Hatamizadeh, D. Terzopoulos, and A. Myronenko. End-to-end boundary aware networks for medical image segmentation. In Inter. Workshop on Machine Learning in Medical Imaging, pages 187–194. Springer, 2019.

[7] E. Hsu, K. Pulli, and J. Popovic. Style translation for human motion. ACM Transactions on Graphics, 24(3):1082–89, 2005.

[8] G. B. Huang, M. Ramesh, T. Berg, and E. Learned-Miller. Labeled faces in the wild: A database for studying face recognition in unconstrained environments. Technical Report 07-49, University of Massachusetts, Amherst, Oct 2007.

[9] Jin Wang, Yu Chen, and M. Adjouadi. A comparative study of multilinear principal component analysis for face recognition. In 2008 37th IEEE Applied Imagery Pattern Recognition Workshop, pages 1–6, 2008.

[10] V. Kazemi and J. Sullivan. One millisecond face alignment with an ensemble of regression trees. In Proc. IEEE Conf. on Computer Vision and Pattern Recognition, CVPR '14, pages 1867–74, Washington, DC, USA, 2014. IEEE Computer Society.

[11] Y. Kim, E. Park, S. Yoo, T. Choi, L. Yang, and D. Shin. Compression of deep convolutional neural networks for fast and low power mobile applications. CoRR, abs/1511.06530, 2015.

[12] D. E. King. Dlib-ml: A machine learning toolkit. Journal of Machine Learning Research, 10:1755–1758, 2009.

[13] V. Lebedev, Y. Ganin, M. Rakhuba, I. V. Oseledets, and V. S. Lempitsky. Speeding-up convolutional neural networks using fine-tuned cp-decomposition. CoRR, abs/1412.6553, 2014.

[14] I. Macedo, E. V. Brazil, and L. Velho. Expression transfer between photographs through multilinear aam's. pages 239–246, Oct 2006.

[15] A. Novikov, D. Podoprikhin, A. Osokin, and D. P. Vetrov. Tensorizing neural networks. In C. Cortes, N. D. Lawrence, D. D. Lee, M. Sugiyama, and R. Garnett, editors, Advances in Neural Information Processing Systems 28, pages 442–450. Curran Associates, Inc., 2015.

[16] I. V. Oseledets. Tensor-train decomposition. SIAM J. on Scientific Computing, 33(5):2295–2317, 2011.

[17] I. V. Oseledets and E. E. Tyrtyshnikov. Breaking the curse of dimensionality, or how to use svd in many dimensions. SIAM J. on Scientific Computing, 31(5):3744–3759, 2009.

[18] A. Shashua and T. Hazan. Non-negative tensor factorization with applications to statistics and computer vision. In Proc. Inter. Conf. on Machine Learning (ICML'05), pages 792–799, New York, NY, 2005. ACM Press.

[19] A. Shashua and A. Levin. Linear image coding for regression and classification using the tensor-rank principle. In Proc. IEEE Conf. on Computer Vision and Pattern Recognition, pages I–42–49, Hawai, 2001.

[20] W. Si, K. Yamaguchi, and M. A. O. Vasilescu. Face tracking with multilinear (tensor) active appearance models. http://pdfs.semanticscholar.org/6c64/59d7cadaa210e3310f3167dc181824fb1bff.pdf, Jun 2013.

[21] L. Sirovich and M. Kirby. Low dimensional procedure for the characterization of human faces. J. of the Optical Society of America A., 4:519–524, 1987.

[22] D. Tao, X. Li, X. Wu, and S. J. Maybank. General tensor discriminant analysis and gabor features for gait recognition. IEEE Transactions on Pattern Analysis and Machine Intelligence, 29(10):1700–1715, 2007.

[23] D. Tao, J. Sun, X. Wu, X. Li, J. Shen, S. J. Maybank, and C. Faloutsos. Probabilistic tensor analysis with akaike and bayesian information criteria. In M. Ishikawa, K. Doya, H. Miyamoto, and T. Yamakawa, editors, Neural Information Processing, pages 791–801, Berlin, Heidelberg, 2008. Springer Berlin Heidelberg.

[24] M. A. Turk and A. P. Pentland. Eigenfaces for recognition. J. of Cognitive Neuroscience, 3(1):71–86, 1991.

[25] M. A. O. Vasilescu. A Multilinear (Tensor) Algebraic Framework for Computer Graphics, Computer Vision, and Machine Learning. PhD thesis, University of Toronto, 2009.

[26] M. A. O. Vasilescu. Multilinear projection for face recognition via canonical decomposition. In Proc. IEEE Inter. Conf. on Automatic Face Gesture Recognition (FG 2011), pages 476–483, Mar 2011.

[27] M. A. O. Vasilescu and E. Kim. Compositional hierarchical tensor factorization: Representing hierarchical intrinsic and extrinsic causal factors. In The 25th ACM SIGKDD Conf. on Knowledge Discovery and Data Mining (KDD19): Tensor Methods for Emerging Data Science Challenges Workshop, Aug. 5 2019.

[28] M. A. O. Vasilescu and D. Terzopoulos. Multilinear projection for appearance-based recognition in the tensor framework. In Proc. 11th IEEE Inter. Conf. on Computer Vision (ICCV'07), pages 1–8, 2007.

[29] D. Vlasic, M. Brand, H. Pfister, and J. Popovic. Face transfer with multilinear models. ACM Transactions on Graphics (TOG), 24(3):426–433, Jul 2005.

[30] H. Wang and N. Ahuja. Compact representation of multidimensional data using tensor rank-one decomposition. In IEEE, Inter. Conf. on Pattern Recognition (ICPR), pages I–44–47, 2004.

[31] H. Wang and N. Ahuja. Rank-R approximation of tensors using image-as-matrix representation. In IEEE Conf. on Computer Vision and Pattern Recognition (CVPR), pages 346–353, 2005.

[32] H. Wang, Q. Wu, L. Shi, Y. Yu, and N. Ahuja. Out-of-core tensor approximation of multi-dimensional matrices of visual data. ACM Transactions on Graphics, 24(3):527–535, 2005.

[33] L. Wang, X. Wang, X. Tong, S. Lin, S. Hu, B. Guo, and H. Shum. View displacement mapping. ACM Trans. Graphics, 22(3):334–339, 2003. Proc. ACM SIGGRAPH 03.

[34] D. Xu, S. Yan, L. Zhang, X. Tang, H. Zhang, and Z. Liu. Concurrent subspaces analysis. In Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR), San Diego, CA, Jun 2005.

[35] J. Yang, X. Gao, D. Zhang, and J. Yang. Kernel ICA: An alternative formulation and its application to face recognition. Pattern Recognition, 38(10):1784–87, 2005.

[36] J. Ye. Generalized low rank approximations of matrices. Machine Learning, 61(1):167–191, 2005.