Character Decompositions

S H Srinivasan
Applied Research Group
Satyam Computer Services Limited
Bangalore - 560 075, INDIA
`SH_Srinivasan@satyam.com`

K R Ramakrishnan
Department of Electrical Engineering
Indian Institute of Science
Bangalore - 560 012, INDIA
`krr@ee.iisc.ernet.in`

Suvrat Budhlakoti
Electronics & Instrumentation
Regional Engg College
Rourkela, India

Abstract

Several decomposition formats exist for image data: principal components, independent components, non-negative components, etc. These decompositions have been applied mainly to natural image data. In this paper, we study the above decompositions for hand-written devanagari character data. We propose a new measure – spatial entropy – for characterizing datasets. Datasets with high spatial entropy are likely to give rise to local features.

1. Introduction

There is a resurgence of interest in techniques which extract “interesting” information from large databases containing images and video. Since the databases are typically unstructured, it is necessary to use unsupervised learning techniques. In particular, unsupervised feature extraction and classification are called for.

Several unsupervised feature extraction techniques exist: principal components, independent components, etc. Most of these techniques have been applied on natural images and the resulting features are shown to be interesting. For example, *eigenfaces* represent prototypical faces, *edges* are independent components, etc. Most of the 2D-datasets on which these techniques are applied are reasonably homogeneous like face datasets. In this paper we apply these techniques to synthetic data: handwritten devanagari characters. Indian language orthography is interesting: there is certain amount of compositionality. Strokes can be composed resulting in complex characters. For example, from the base character क, we can derive characters का, कि, की, etc. We can decompose any character to the basic strokes it contains. Our goal is to identify decompositions which correspond to this intuition.

This paper is organized as follows. Section 2 describes the decompositions used in this paper. Section 3 describes the results of experiments. Section 4 concludes the paper.

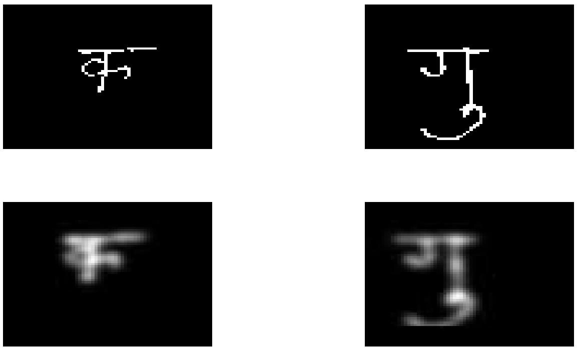


Figure 1: Sample characters. The top row contains scanned inverted-binary 64×64 examples and the bottom contains the corresponding Gaussian-smoothed gray scale versions.

2. Matrix decompositions

Many datasets can be represented as matrices. Consider a collection of images having the same size. This can be represented as a matrix: the rows (or columns) represent each image. There are as many rows (or columns) as the number of images. This matrix can be subjected to several matrix decompositions. The results of the decompositions are then interpreted in terms of the problem domain. The most popular decomposition is the eigen decomposition which is discussed below.

2.1 Eigen decomposition

The covariance matrix of the data matrix is subjected to eigen value decomposition. Let $x_1, x_2, \cdots, x_n$ be the data vectors. The covariance matrix, C, is defined as

$$C = \frac{1}{n} \sum (x_i - \mu)(x_i - \mu)^T$$

where $\mu = (1/n) \sum x_i$ is the mean value of the vectors. We now look for bases which diagonalize C (or decorrelate the vectors $x_1, x_2, \cdots, x_n$). This is given by

$$C = U^T \Lambda U$$

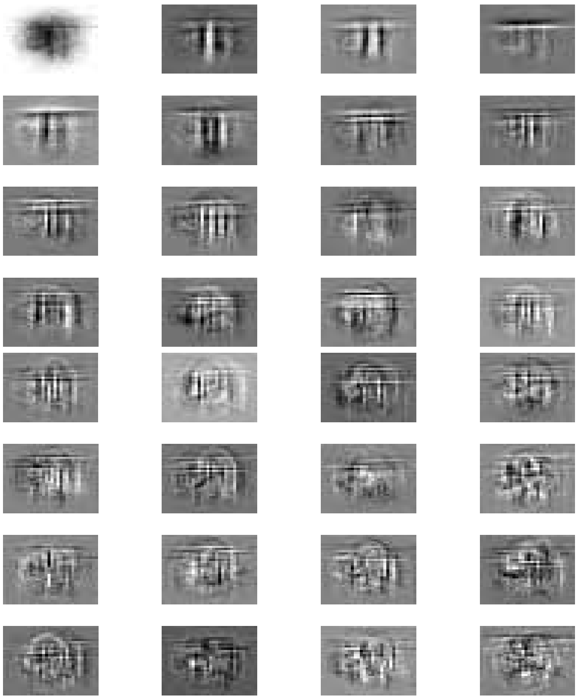


Figure 2: Eigen components of क-family and ग-family

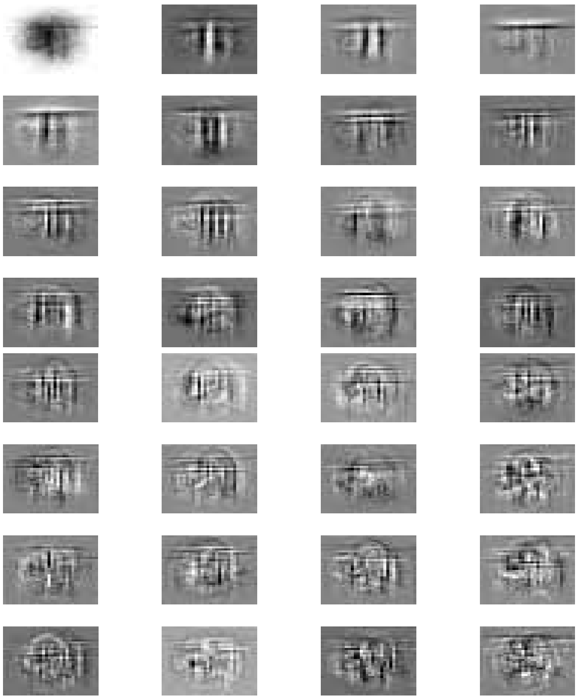


Figure 3: Singular components of क-family and ग-family

The columns of U correspond to the eigenvectors of C and the diagonal entries of Λ correspond to the eigen values. The eigenvectors are orthonormal. The eigenvectors are ordered according to their importance - as indicated by the associated eigenvalues.

Eigen decomposition has been applied to face images [8]. The eigenfaces (eigen vectors of the covariance matrix of face images) correspond to prototypical faces.

2.2 Singular value decomposition

While eigen decomposition works on the covariance matrix, SVD work with the data matrix itself. The $m \times n$ data matrix X is decomposed as

$$X = USV^T$$

where U is $m \times m$, V is $n \times n$, and S is $m \times n$. The diagonal entries of S are known as singular values. (While eigen decomposition is defined only for square matrices, SVD is defined for rectangular matrices too. SVD is a generalization of eigen decomposition.) The columns of U corresponding to the singular values form a basis for the column space of X and those of V form the basis for the row space of X. If the input images form the rows of X, then the columns of V form a basis of that space. These basis vectors can also be ordered according to their "importance" as given by the singular values. Singular value decompositions are very popular in document analysis [1].

2.3 Independent Component Analysis

Independent Component Analysis (ICA) is a generalization of stochastic interpretation of matrix diagonalization. Diagonalization, when applied to stochastic vectors, produces mutually uncorrelated basis. ICA produces a *statistically independent* basis. Thus the independent components of a matrix X can be thought of a collection of statistically independent sources for the rows (or columns) of X [5]. The decomposition reveals the sources as well as mixing coefficients. The $m \times n$ matrix X is decomposed as

$$X = WS + N$$

where S is the $r \times n$ source signal matrix, W is the $m \times r$ mixing matrix, and N is the matrix of noise signals. Here r is the number of independent sources. The above decomposition can be performed for any number of independent components and the sizes of W and S vary accordingly. We use the Fast ICA algorithm for performing the decomposition [2].

2.4 Non-negative Matrix Factorization

All the above decompositions have a problem of interpretation. Usually the data matrix has only positive values.

But the components extracted can have negative entries. It is difficult to interpret negative entries. In particular, we cannot interpret them as *parts* of objects like documents. Non-negative matrix factorization (NMF) [3] [4] attempts a factorization in which the components have non-negative entries. The NMF of X is given by

$$X = WH$$

where the factors W and H contain only non-negative entries. The interpretation in [3] for the above decomposition is as follows: The columns of the $m \times n$ matrix X are the signals, the columns of the $m \times r$ matrix W are the basis signals, and the $r \times n$ matrix H is the mixing matrix. (Here r is the number of parts or non-negative components.) We can use the columns of W as the new basis. NMF has been applied on face datasets and it has been shown that they learn parts of faces [3].

2.5 Local NMF

Though it is claimed that NMF produces localized parts of faces in [3], in the examples shown in [6], they produce holistic representations. It is desirable to have decompositions which are guaranteed to produce more local representations. Hence [6] proposes a constrained NMF which is called LNMF. The constraints on the basis vectors are

1. The basis vectors should have as many non-zero entries as possible and orthogonal.

2. The components capture as much information as possible about the input vectors.

LNMF has been shown to produce localized representations for face datasets. We show that for certain datasets LNMF also produces global features (section 3.3). We resolve the paradox by proposing a measure - spatial entropy – which characterizes which datasets are likely to produce local representations (section 4).

3. Results

We first created a database containing handwritten devanagari characters. The database contained the characters from क, ग, च, ज, ट, ड, त, द, प, ब, य, र, ल, and व families. क-family, for example, consists of the following characters: क, का, कि, की, कु, and कू. Samples were created from 28 persons. Hence the database has 2352 characters. The characters are available as 64 × 64 bitmaps. The characters were inverted (white-on-black from black-on-white) before processing.

We also created a derived dataset by smoothing images using a 2D Gaussian mask. Some sample characters from the database are shown in figure 1.

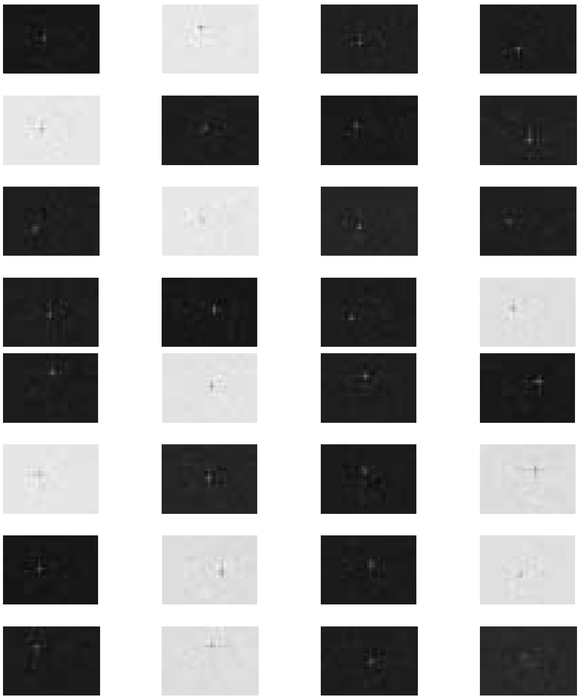


Figure 4: Independent components of क-family and ग-family

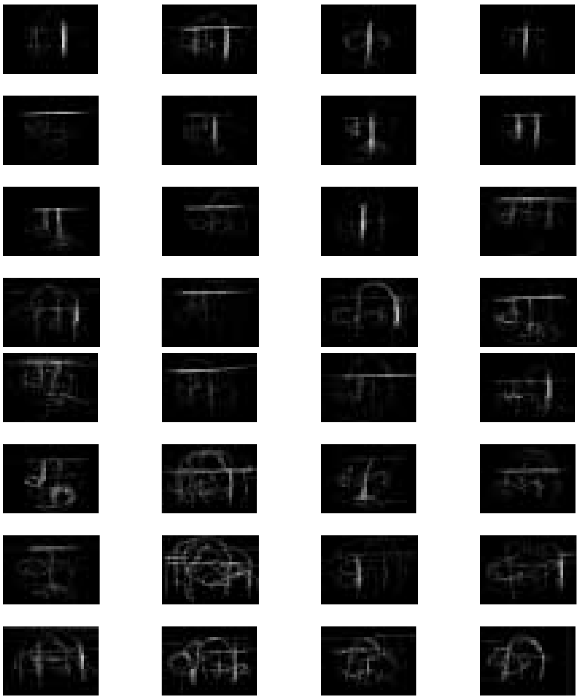


Figure 5: Non-negative factors of क-family and ग-family

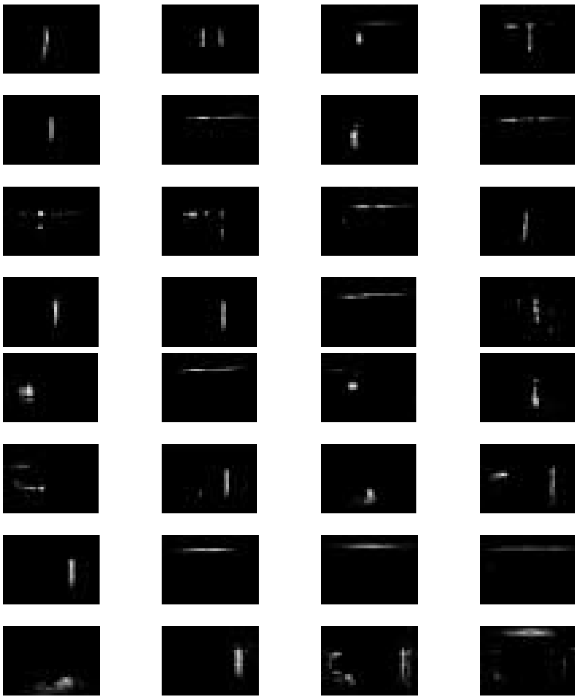


Figure 6: LNMF of क-family and ग-family

3.1 क - ग family decompositions

The above five decompositions were performed on क and ग-families. 32-components were extracted in all cases. The results are displayed based on the following notions.

1. As mentioned before, some decompositions produce components containing negative entries. For visualization, these components were transformed to the pixel value range (0-255) by subtracting the minimum value in the component and then dividing by the range.

2. ED, SVD, and ICA produce components which can be ordered according to their importance. For NMF, we use the cumulative contribution of components to rank them. The contributions can be obtained from the H matrix.

The resulting components are shown in figures 2 to 6. Figures 2 and 3 confirm the well-known feature of eigen and singular value decompositions: they produce global representation of the prototypes. For example, it is possible to identify क and ग in the figures. ICA representations (figure 4) can not be readily interpreted in our case. The seem to high light end points and intersections. NMF basis functions (figure 5) are holistic. They correspond to input characters with some noise. LMNF basis functions (figure 6) are the closest to our intuitive idea of strokes. But there is an absence of curves: straight line features are emphasized.

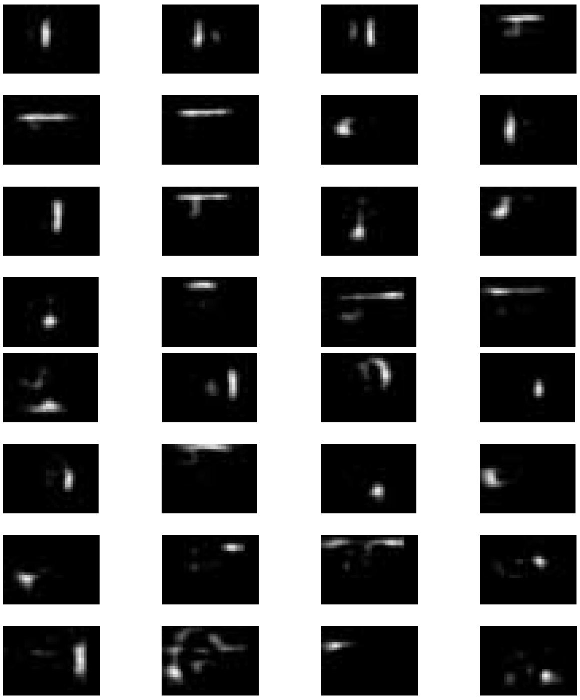


Figure 7: Non-negative factors of smoothed क-family and ग-family

3.2 Smoothing

To extract common features in images, similar regions (like similar strokes) should spatially overlap across images. Since the scanned images have very thin lines, such overlap is unlikely across images. If the images are smoothed, such overlaps are more likely. Hence we subjected the smoothed images to the above decompositions. There was no qualitative difference in the behavior of ED, SVD, and ICA in these cases. The results for NMF and LNMF are shown in figures 7 and 8. It can be seen that NMF is able to extract straight lines as well as curves to certain extent. In contrast, LNMF features are more localized.

3.3 टि-जी features

In the above experiments, there is a paucity of curves in extracted features. So we decided to extract features only for the two characters: टि and जी. These two characters have lot of curved strokes. Eight basis vectors were extracted for NMF and LNMF. (Other decompositions were not performed.) The results are shown in figure 9. It can be seen that LNMF extracts stroke-like features only for smoothed images. For other images, the features are holistic.

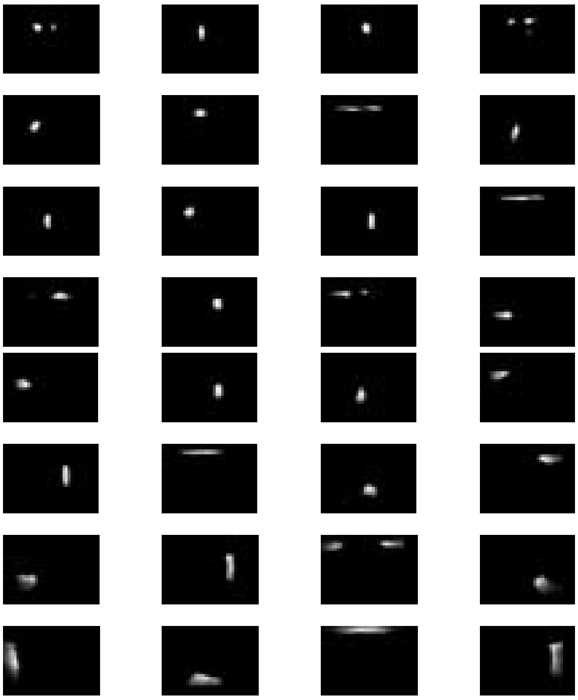


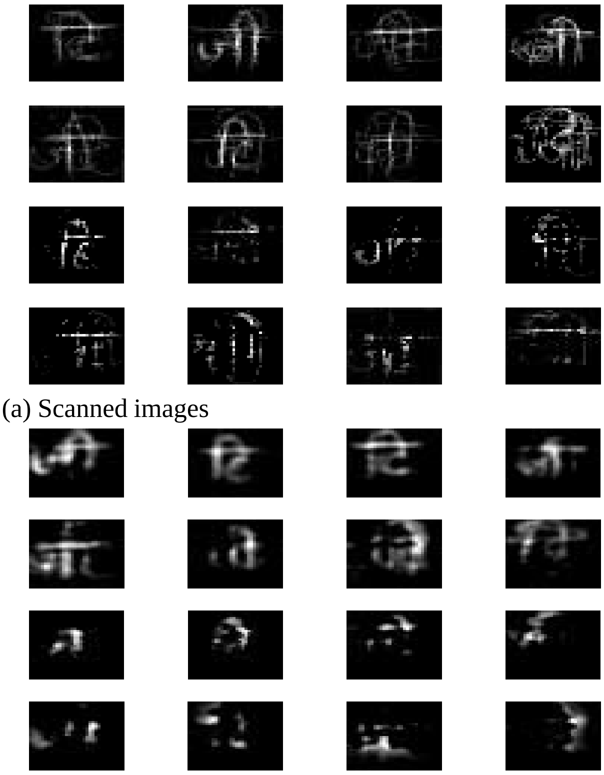
Figure 8: LNMF of smoothed क-family and ग-family

4. Discussion

We have performed several linear decompositions on devanagari characters. NMF and LNMF are better at producing part-like decompositions. The behavior of these algorithms depends heavily on the nature of the dataset. Smoothed datasets are more likely to produce part-like features. It will be useful if we can characterize a dataset's potential for producing local representations – *before* subjecting it to decompositions.

We now present a quantitative measure – spatial entropy – which characterizes the datasets which are likely to produce local part-like representation. We define spatial entropy as follows. Consider a dataset which is represented as a matrix with the rows corresponding to images. The columns correspond to pixels. We can calculate the probability mass function for each pixel value and hence the entropy for each pixel. We define the spatial entropy as the sum of pixel entropies. If the spatial entropy of the dataset is small, the dataset is likely to produce global, holistic features. If the spatial entropy is large, then the dataset is likely to produce local, part-like features.

To test the effectiveness of this measure, we randomly selected varying number of images from the database and calculated the spatial entropy. The number of images and the corresponding spatial entropies are shown in figure 10. It can be seen that the spatial entropy increases with the number of images. (The entropy calculated without con-



(a) Scanned images

(b) Smoothed images

Figure 9: Decompositions of टि and जी. Eight features were extracted in each decomposition. For (a), the scanned images are used. The top two rows correspond to NMF and the next two rows correspond to LNMF. For (b), smoothed images are used. Here also, the top two rows correspond to NMF and the next two rows correspond to LNMF.

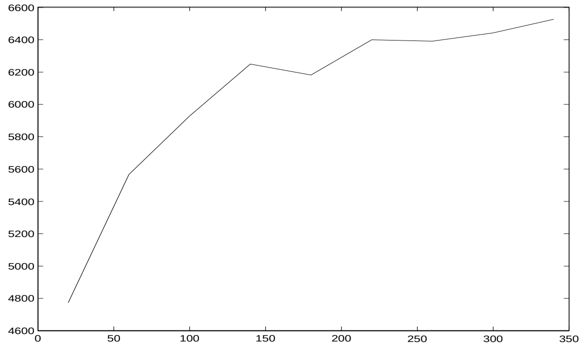


Figure 10: Spatial entropy vs. data set size. The images were randomly chosen. Spatial entropy increases with dataset size.

sidering the spatial information does not increase with the number of images.) Datasets with low entropy produce global representations and datasets with high entropy produce global representations.

We also calculated the spatial entropy of smoothed datasets. In this case, the characters used were from the क and प family. The *same set of characters* were subjected three levels of smoothing. The spatial entropy values for the resulting datasets were: 3923, 6883 and 10038. Here also, higher spatial entropy resulted in localized representations (data not shown here).

5 Conclusions

In this paper, we studied the decompositions of characters by various techniques. NMF and LNMF produce part-like representations when the datasets have "high" spatial entropy. The spatial entropy thresholds above which local representations emerge has to be calculated. It is likely that this threshold is higher for NMF. Also, calculation of this threshold for different datasets (faces, logos, etc.) is will be an interesting extension of the work reported here.

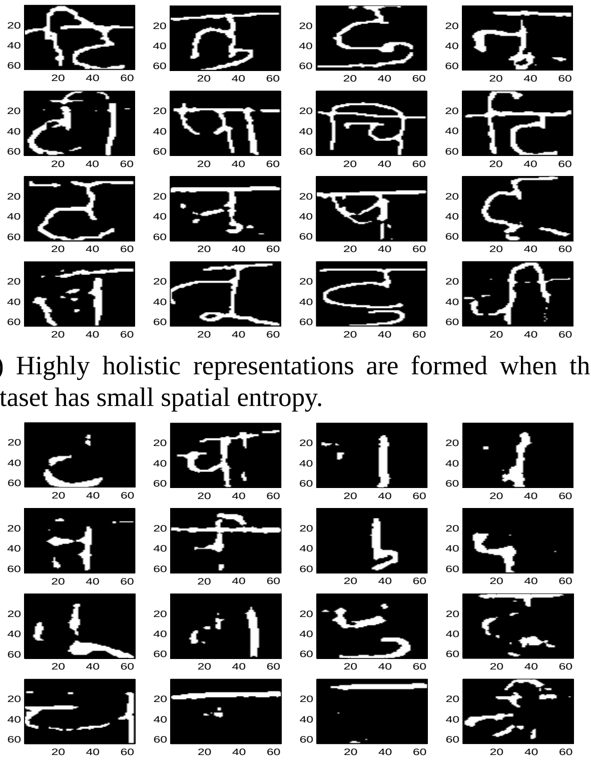
The decompositions considered here are linear. It will also be interesting to extend the decompositions to nonlinear transforms [7].

References

⟦ 1 ⟧ S. T. Dumais, G. W. Furnas, T. K. Landauer, and S. Deerwester. Using latent semantic analysis to improve information retrieval. In *Proceedings of CHI'88*, 1988.

⟦ 2 ⟧ A. Hyvarinen. Fast and robust fixed-point algorithms for indepdendent component analysis. *IEEE Transactions on Neural Networks*, 10(3):626–634, 1999.

⟦ 3 ⟧ D. D. Lee and H. S. Seung. Learning the parts of objects by non-negative matrix factorization. *Nature*, 401:788–791, 1999.



(a) Highly holistic representations are formed when the dataset has small spatial entropy.

(b) Part like representations emerge when the spatial entropy of the dataset is large.

Figure 11: NMF decompositions of randomly chosen character data. (a) Size of dataset = 30 (b) Size of dataset = 336.

⟦ 4 ⟧ D. D. Lee and H. S. Seung. Algorithms for non-negative matrix factorization. In *Proc. NIPS*, 2000.

⟦ 5 ⟧ T.-W. Lee, M. Girolami, A. J. Bell, and T. J. Sejnowski. A unifying information-theoretic framework for independent component analysis. *International Journal on Mathematical and Computer Modeling*, 1998.

⟦ 6 ⟧ S. Z. Li, X. Hou, H. Zhang, and Q. Cheng. Learning spatially localized, parts-based representation. In *Proc. CVPR*, 2001.

⟦ 7 ⟧ B. Scholkopf, A. Smola, and K.-R. Muller. Nonlinear component analysis as a kernel eigenvalue problem. *Neural Computation*, 10, 1998.

⟦ 8 ⟧ M. A. Turk and A. P. Pentland. Face recognition using eigenfaces. In *CVPR*, 1991.