Deep Grouped Non-Negative Matrix Factorization Method for Image Data Representation

Zihao Zhan¹, Wen-Sheng Chen¹,²,³, Binbin Pan¹,²,³, Bo Chen¹,²,³

¹College of Mathematics and Statistics, Shenzhen University, China
²Guangdong Key Laboratory of Media Security, Shenzhen University, China
³Shenzhen Key Laboratory of Advanced Machine Learning and Applications, Shenzhen University, China

E-MAIL: {chenws,pbb,chenbo}@szu.edu.cn
Corresponding Author: Wen-Sheng Chen

Abstract:
Non-negative matrix factorization (NMF) is an unsupervised learning method that can be exploited for parts-based image representation due to non-negativity constraints. However, single-layer NMF cannot capture the latent hierarchical structure features from images, while deep features play more important roles in the image representation and recognition tasks. To overcome the limitation of NMF, this paper proposes a novel deep grouped NMF (DGNMF) approach to learn different level attributes of the data. It is interesting that DGNMF approach automatically makes the data from distinct classes share different basis images and the feature vectors among different classes are mutually orthogonal at the same layer. Meanwhile, to preserve the local information, our DGNMF model establishes the objective function with graph regularization, and its optimization problem is solved using gradient descent method. The developed DGNMF algorithm is proved to be convergent and is finally evaluated on face datasets for classification. Compared with some state-of-the-art deep NMF variants, the results demonstrate the proposed DGNMF algorithm achieves surpassing performances using different layer features.

Keywords:
Deep Non-Negative Matrix Factorization; Supervised Learning Method; Image Data Representation

1. Introduction

Finding a meaningful data representation is a central problem of pattern recognition and machine learning. Non-negative matrix factorization [1] [2] is a low-rank decomposition method for parts-based image representation. It approximately uses the product of two non-negative matrices, namely a basis matrix and a coefficient matrix, to express a non-negative data matrix. NMF adopts a single layer decomposition to acquire the shallow attributes of the data and thus cannot disclose the underlying hierarchical information from the complex data. Deep learning technologies [4] [5] have shown that hierarchical features are significant in image representation and pattern recognition. Therefore, deep non-negative matrix factorization (DNMF) variants [6]- [10], also known as multi-layer NMF, have been developed to improve the single-layer NMF algorithms. DNMF approaches are mainly categorized into two classes: coefficient matrix-based decomposition and basis matrix-based decomposition. The coefficient matrix-based DNMF methods [6]- [8] repeatedly decompose the coefficient matrix to obtain the multi-layer decomposition, while the basis matrix-based DNMF methods [9] [10] continuously factorize the basis matrix to generate the deep structure. Ahn et al. proposed a multi-layer non-negative matrix factorization (MNMF) [2] using NMF with KL-divergence, which is successfully applied to dynamic pet image analysis. Cichocki et al. employed different cost functions, such as generalized KL-divergence or Euclidean distance, to present multi-layer NMF algorithms with multi-start initialization for blind source separation [7] [8]. To improve semi-NMF [3], Trigeorgis et al. came up with a deep semi-NMF (DSNMF) method [9] suitable for clustering of multi-modally distributed objects such as facial images. Zhao et al. gave a framework of DNMF architecture based on underlying basis image learning (RDNBMF) [10], which is advantageous to the representation of the facial images and achieves better performance in face recognition tasks. However, the NMF and aforementioned DNMF approaches do not utilize the knowledge of class labels and thus are unsupervised learning methods. It is known that label knowledge is more beneficial to the improvements of the unsupervised algorithms. Also, most DNMF algorithms do not
discuss their convergence of deep decomposition. They cannot guarantee the reasonableness of the algorithms. To address the above-mentioned problems, this paper proposes a novel deep grouped NMF (DGNMF) approach. The proposed DGNMF regards the data from the same class as a group to recursively perform NMF on the basis-image matrices and assemble all of the grouped multi-layer factorizations to yield the deep decomposition of the whole training data. It automatically leads to the image data from different classes having different basis images and the feature vectors from different classes are orthogonal to each other at each layer. These properties are conducive to enhancing the discriminating power of the proposed algorithm. We further incorporate the graph regularization into the cost function to keep the locality of the data and theoretically show the convergence of the DGNMF algorithm using auxiliary function technique. Two publicly available face datasets, namely Yale dataset and FERET dataset, are used to assess the performances of the compared deep NMF algorithms. The results indicate the effectiveness and better accuracy of the proposed DGNMF algorithm under different decomposition layers.

2. Related Works

This section will give a brief introduction to MNMF [6] and DNBMF [10].

2.1. MNMF

Let \( X = [x_1, x_2, \ldots, x_n] \in R_+^{m \times n} \) be a non-negative data matrix. MNMF decomposes the coefficient matrix hierarchically to yield a multi-layer structure and gets its deep factorization as \( X \approx W_1 W_2 \cdots W_L H_L \), where \( L \) is the layer number as the deep NMF architecture. The update rule of the \( l \)th layer are as follows:

\[
\begin{align*}
H_l^{(t+1)} &= H_l^{(t)} \otimes (W_l^{(t)} H_{l-1}^{(t)}) \odot (W_l^{(t)} W_l^{(t)} H_l^{(t)}), \\
W_l^{(t+1)} &= W_l^{(t)} \odot (H_{l-1} H_l^{(t)}) \odot (W_l^{(t)} H_l^{(t)} H_l^{(t)}), \\
\end{align*}
\]

(1)

where \( \otimes \) and \( \odot \) denote the element-wise multiplication and division, respectively.

2.2. DNBMF

Unlike MNMF, DNBMF decomposes the basis matrix repeatedly to obtain a multi-layer structure and the final factorization is \( X \approx W_L H_L \cdots H_2 H_1 \). The update rule of the \( l \)th layer are as below:

\[
\begin{align*}
H_l^{(t+1)} &= H_l^{(t)} \otimes (W_l^{(t)} W_l^{(t)} H_l^{(t)}), \\
W_l^{(t+1)} &= W_l^{(t)} \odot (H_{l-1} H_l^{(t)}) \odot (W_l^{(t)} H_l^{(t)} H_l^{(t)}) .
\end{align*}
\]

(2)

3. Deep Grouped Non-Negative Matrix Factorization

3.1. DGNMF model

Let the image data matrix \( X \) with \( c \) classes be \( X = [X_1, X_2, \ldots, X_c] \), where the data from the \( i \)-th class are grouped into the sub-matrix \( X_i = [x_1^{(i)}, x_2^{(i)}, \ldots, x_n^{(i)}] \in R_+^{m \times n_i} \), \( i = 1, 2, \ldots, c \) and the total training number \( n = \sum_{i=1}^{c} n_i \).

For the first layer, we conduct NMF on \( c \) grouped sub-matrices \( X_i \) (\( i = 1, 2, \ldots, c \)) and obtain the decomposition \( X_i \approx W_i^{(1)} H_i^{(1)} \), where \( W_i^{(1)} \in R_+^{m \times r_{i1}} \) is the sub-basis matrix of the \( i \)th class, and \( H_i^{(1)} \in R_+^{r_{i1} \times n_i} \) is the corresponding sub-feature matrix. We see that the factorization on the entire data matrix \( X \) can be written as

\[
X \approx W_1 H_1 , \quad (3)
\]

where \( W_1 = [W_1^{(1)}, W_2^{(1)}, \ldots, W_c^{(1)}] \in R_+^{m \times r_1} \), \( H_1 = \text{mdiag}[H_1^{(1)}, H_2^{(1)}, \ldots, H_c^{(1)}] \in R_+^{r_1 \times n} \) and \( r_1 = c r_1 \). For the \( l \)th layer factorization, we perform NMF on \( W_l^{(l-1)} \) and have \( W_l^{(l-1)} \approx W_i^{(l)} H_i^{(l)} \), where the sub-basis matrices \( W_i^{(l-1)} \in R_+^{m \times r_{i1}} \), \( W_i^{(l)} \in R_+^{m \times r_l} \) and the sub-feature matrices \( H_i^{(l)} \in R_+^{r_l \times n_{il}} \), the number of sub-features \( r_l = c r_l \), \( i = 1, 2, \ldots, c \) and \( l = 2, 3, \ldots, L \). Hence, the factorization of the \( l \)th layer can be formulated as

\[
W_{l-1} \approx W_l H_l , \quad (4)
\]

where \( W_{l-1} = [W_1^{(l-1)}, W_2^{(l-1)}, \ldots, W_c^{(l-1)}] \in R_+^{m \times r_{l-1}} \), \( W_l = [W_1^{(l)}, W_2^{(l)}, \ldots, W_c^{(l)}] \in R_+^{m \times r_l} \) and \( H_l = \text{mdiag}[H_1^{(l)}, H_2^{(l)}, \ldots, H_c^{(l)}] \in R_+^{r_l \times n_{il}} \). If denote \( \tilde{H}_l \) and \( \tilde{H}_l \) respectively by

\[
\tilde{H}_l = H_l H_{l-1} \cdots H_1 \quad \text{and} \quad \tilde{H}_l^{(1)} = H_l \cdots H_2 H_1^{(1)} , \quad (5)
\]

then \( \tilde{H}_l = \text{mdiag}[\tilde{H}_1^{(l)}, \tilde{H}_2^{(l)}, \ldots, \tilde{H}_c^{(l)}] \) and it recursively yields the final deep decomposition as

\[
X \approx W_L \tilde{H}_L , \quad (6)
\]

where \( W_L = [W_1^{(L)}, W_2^{(L)}, \ldots, W_c^{(L)}] \) is the basis matrix at layer \( L \), and \( W_L^{(L)} \) is the underlying sub-basis matrix of the sample matrix \( X_i \) from class \( i \). The feature matrix \( \tilde{H}_L \) can be expressed at layer \( L \) as follows:

\[
\tilde{H}_L = \begin{bmatrix} 0 & 0 & \cdots & 0 \\ 0 & \tilde{H}_2^{(L)} & \cdots & 0 \\ 0 & \ddots & \ddots & \ddots \\ 0 & \cdots & \cdots & \tilde{H}_c^{(L)} \end{bmatrix} , \quad (7)
\]
It can be seen from (6) and (7) that at each layer the data from different classes share different basis images and the feature vectors among distinct classes are mutually orthogonal. To further preserve the locality structure of the data, we will introduce an adjacency matrix \( S = (S_{mn})_{n \times n} \) with its entries defined by

\[
S_{mn} = \begin{cases} 
1 & \text{if } x^i_m \in N_k(x^i_n) \text{ or } x^i_n \in N_k(x^i_m), \\
0 & \text{else},
\end{cases}
\]

where notation \( N_k(x) \) means the \( k \) nearest neighbors of data \( x \) and \( g(x^i_m, x^i_n) = \exp(-\|x^i_m - x^i_n\|^2/\tau^2) \) with parameter \( \tau > 0 \).

At the \( l \)th layer, let \( a^l_m \) be the \( m \)th column of feature matrix \( H^l \) defined by (5) and \( Q_l \) be the local scatter quantity given by

\[
Q_l = \frac{1}{2} \sum_{m,n=1}^{n} \|a^l_m - a^l_n\|^2 S_{mn} = Tr(H^l L^l H^l)^	op,
\]

where \( L = D - S \) is the Laplacian matrix, \( Tr(\cdot) \) is the trace of a matrix and \( D \) is a diagonal matrix with elements \( D_{nm} = \sum_n S_{mn} \). We can easily derive that \( Q_l = \sum_{i=1}^{c} Tr(H_i^l (L^l H_i^l)^	op) \), where matrix \( L^l = D^l - S^l \) is a Laplacian matrix of class \( i \), which is generated by the adjacency matrix \( S^l = (S^l_{mn})_{n \times n} \), of class \( i \) and its elements defined by

\[
S^l_{mn} = \begin{cases} 
1 & \text{if } x^i_m \in N_k(x^i_n) \text{ or } x^i_n \in N_k(x^i_m), \\
0 & \text{else},
\end{cases}
\]

and \( D^l = \text{diag}(D^l_{mn}) \) with \( D^l_{nm} = \sum_n S^l_{mn} \).

Based on above analysis, the objective function of our DGNMF method at layer \( l \) can be established as follows:

\[
F_l(W_l, H_l) = \frac{1}{2} \sum_{i=1}^{c} \|W_i^{l(i)} - W_i^{2(i)} H_i^l\|_2^2 + \frac{\alpha}{2c} Q_l,
\]

where \( \alpha > 0 \) and it can be reformulated as:

\[
F_l(W_l, H_l) = \frac{1}{c} \sum_{i=1}^{c} F_i^{(l)}(W_i^{(l)}, H_i^{(l)}),
\]

where

\[
F_i^{(l)}(W_i^{(l)}, H_i^{(l)}) = \frac{1}{2} \|W_i^{(l-1)} - W_i^{(l)} H_i^{(l)}\|_2^2 + \frac{\alpha}{2c} Tr(H_i^{l(i)} L^l H_i^{l(i)})^	op.
\]

The optimization problem of DGNMF is then as below:

\[
\min_{W_i \geq 0, H_i \geq 0} F_i(W_i, H_i).
\]

Above problem (14) is equivalent to the following problem:

\[
\min_{W_i \geq 0, H_i \geq 0} F_i^{(l)}(W_i^{(l)}, H_i^{(l)}), \quad i = 1, 2, \ldots, c.
\]

The problem (15) can be solved using gradient descent method and the update rules of the proposed DGNMF method are acquired as:

\[
H_i^{(l)} \leftarrow H_i^{(l)} \odot (W_i^{(l)\top} W_i^{(l-1)} + \alpha H_i^{(l)\top} H_i^{(l-1)} S^l H_i^{(l-1)\top} + (W_i^{(l)\top} W_i^{(l-1)} H_i^{(l-1)} + \alpha H_i^{(l)\top} H_i^{(l-1)} D^l H_i^{(l-1)\top}),
\]

\[
W_i^{(l)} \leftarrow (W_i^{(l)} \odot W_i^{(l-1)} H_i^{(l-1)\top}) \odot W_i^{(l)} H_i^{(l)} H_i^{(l-1)\top}.
\]

### 3.2. Convergence Analysis

This subsection will construct an auxiliary function to prove the convergence of the proposed DGNMF algorithm.

**Definition:** \( G(h, h^{(l)}) \) is called an auxiliary function of \( F(h) \), if for all \( h, h^{(l)} \in \mathbb{R}^n \), there are \( G(h, h^{(l)}) \geq F(h) \) and \( G(h, h) = F(h) \).

**Lemma 1.** \( F(h) \) is non-increasing under the recursion \( h^{l+1} = \arg \min_{h} G(h, h^{(l)}) \), where \( G(h, h^{(l)}) \) is an auxiliary function for \( F(h) \).

Assume \( w_q^{(l-1)} \) and \( h_q^{(l)} \) are the \( q \)th columns of \( W_i^{(l-1)} \) and \( H_i^{(l)} \) respectively. The following objective function \( f(h_q^{(l)}) \) just takes \( h_q^{(l)} \) as variable:

\[
f(h_q^{(l)}) = \frac{1}{2} \|w_q^{(l-1)} - W_q^{(l)} h_q^{(l)}\|_2^2 + \frac{\alpha}{2c} Tr(H_q^{l(q)} L^{(q)} H_q^{l(q)})^	op.
\]

**Theorem 1.**

\[
G(h_q^{(l)}, h_q^{(l)}) = f(h_q^{(l)}) + (h_q^{(l)} - h_q^{(l-1)})^\top N(h_q^{(l)}) (h_q^{(l)} - h_q^{(l-1)}),
\]

where \( N(h_q^{(l)}) \) is a diagonal matrix with

\[
[N(h_q^{(l)}(i))_{kk}] = [W_i^{(l-1)\top} W_i^{(l)} h_q^{(l)}(k)] \odot [h_q^{(l-1)}(k)] + [\alpha H_i^{(l-1)\top} D^l H_i^{(l-1)\top} q] \odot [h_q^{(l)}(k)]_k,
\]

then \( G(h_q^{(l)}, h_q^{(l)}) \) is the auxiliary function of \( f(h_q^{(l)}) \).

**Theorem 2.** For fixed \( H_i^{(l-1)} \), \( W_i^{(l)} \) and \( h_q^{(l)} \) (\( p \neq q \)), we have that \( f(h_q^{(l)}) \) is non-increasing under the following update rule:

\[
h_q^{(l)} \leftarrow h_q^{(l)} \odot (W_i^{(l)\top} W_i^{(l-1)} q + \alpha H_i^{(l)\top} H_i^{(l-1)} S^l H_i^{(l-1)\top} q) \odot (W_i^{(l)\top} W_i^{(l-1)} q + \alpha H_i^{(l)\top} H_i^{(l-1)} D^l H_i^{(l-1)\top} q).
\]
Step 2. According to (10), calculate $S$.

Step 4. If the inverse of the basis matrix $W$ is computed by (5). Update $H_i^{(l)}$ and $W_i^{(l)}$ according to (16) and (17), respectively.

Step 5. Output $H_1, H_2, \cdots, H_L$ and $W_L$.

For a testing sample $y$, its attribute at layer $l$ can be calculated via $h_i^l = W_i^{-1} y$, where $W_i^{-1}$ is the Moore-Penrose pseudo inverse of the basis matrix $W_i$.

If $s = \arg\min_j \| h_i^l - \tilde{H}_j \|_F$, then $y$ is assigned to class $s$, where $\tilde{H}_j$ is the mean feature vector of $X_j$, which represents the centroid of the $j$th class in the $l$th layer feature space.

4. Experimental Results

This section will report the performance of the proposed DGNMF approach on face recognition. Two face databases, namely Yale database and FERET database, are selected for evaluations. The state-of-the-art deep NMF algorithms, such as MNMF [6], DSNMF [9] and RDNBMF [10], are chosen for comparisons under the same condition.

4.1. Face Datasets

Yale dataset, created by the Yale center for computational vision and control, consists of 165 images of 15 people, each with 11 images of human faces with different expressions, postures and lighting conditions. Each image is with a size of 100×100.

FERET dataset comprises 720 facial images of 120 subjects. Each person has 6 images with different illumination conditions, facial expressions, poses and ages. The resolution of each image is 112×92.

4.2. Comparisons using Deep Feature

This subsection will consider using the second layer feature for comparisons. The experimental settings are as follows. The maximum number of iterations is $I_{\text{max}} = 500$. The number of basis images of all the compared algorithms at layer 2 are opted according to the best algorithmic performance. On the Yale database, we select the parameters of our method as: $r^1 = 5$, $r^2 = 10$, $t = 1000$, and $\alpha = 0.001$. The parameters $(r^1, r^2)$ of MNMF, DSNMF and RDNBMF are set to $(160, 180)$, $(150, 180)$, and $(120, 140)$, respectively. While for the FERET dataset, the parameters of our DGNMF method are assigned as: $r^1 = 10$, $r^2 = 15$, $t = 1000$, and $\alpha = 0.01$. The parameters $(r^1, r^2)$ of MNMF, DSNMF and RDNBMF are given as $(120, 140)$, $(110, 130)$, and $(140, 160)$, respectively.

For the Yale database, we randomly select $TN$ ($TN = 2, 3, \cdots, 9$) images from each person for training and the rest of the $(11 - TN) \times 15$ images are used for testing. Each experiment is run ten times and the average results are tabulated in Table1 and plotted in Fig.1. It can be seen from Table 1 that the accuracies of all compared algorithms ascend with the increase of training samples. Among the compared deep NMF algorithms, the proposed algorithm achieves the best performance, while DSNMF is the worst on Yale dataset. RDNBMF is slightly better than MNMF. More interesting, the basis matrix decomposition-based algorithms, such as DGNMF and RDNBMF, outperform the feature matrix decomposition-based algorithms including MNMF and DSNMF. The results may be due to the regularization constraints. Different from DGNMF and RDNBMF methods, MNMF and DSNMF have no regularization constraints imposed on basis matrix or feature matrix. Especially, our DGNMF is a supervised learning method and the rest compared deep NMF algorithms are unsupervised methods. It also indicates that supervised learning methods outperform unsupervised methods in pattern recognition.

The experiments on the FERET dataset are similar to that of Yale dataset. We randomly opt $TN$ ($TN = 2, 3, 4, 5$) images from each person for training, and the rest of the $(6 - TN) \times 120$ images are for testing. The mean results are recorded in Table2 and plotted in Fig.2. It can be seen that the proposed DGNMF method also achieves excellent performance, which has nearly 10% improvement over MNMF, the worst method on FERET.
TABLE 1. Mean accuracy (%) versus Training Number (TN) on Yale database with $L = 2$

<table>
<thead>
<tr>
<th>TN</th>
<th>MNMF</th>
<th>(std)</th>
<th>DSNMF</th>
<th>(std)</th>
<th>RDNBMF</th>
<th>(std)</th>
<th>DGNMF</th>
<th>(std)</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>73.21</td>
<td>(5.62)</td>
<td>75.12</td>
<td>(3.51)</td>
<td>75.57</td>
<td>(4.00)</td>
<td>85.21</td>
<td>(2.20)</td>
</tr>
<tr>
<td>3</td>
<td>83.11</td>
<td>(3.97)</td>
<td>81.01</td>
<td>(2.27)</td>
<td>85.32</td>
<td>(3.85)</td>
<td>93.11</td>
<td>(1.94)</td>
</tr>
<tr>
<td>4</td>
<td>88.57</td>
<td>(3.17)</td>
<td>86.21</td>
<td>(3.54)</td>
<td>89.11</td>
<td>(3.22)</td>
<td>95.27</td>
<td>(1.39)</td>
</tr>
<tr>
<td>5</td>
<td>91.21</td>
<td>(3.95)</td>
<td>87.11</td>
<td>(4.56)</td>
<td>90.23</td>
<td>(3.82)</td>
<td>95.27</td>
<td>(3.32)</td>
</tr>
<tr>
<td>6</td>
<td>91.58</td>
<td>(5.45)</td>
<td>90.21</td>
<td>(3.06)</td>
<td>94.11</td>
<td>(3.31)</td>
<td>94.11</td>
<td>(4.59)</td>
</tr>
<tr>
<td>7</td>
<td>93.78</td>
<td>(3.27)</td>
<td>91.21</td>
<td>(4.66)</td>
<td>94.12</td>
<td>(3.69)</td>
<td>94.11</td>
<td>(4.56)</td>
</tr>
<tr>
<td>8</td>
<td>94.89</td>
<td>(5.83)</td>
<td>92.32</td>
<td>(3.31)</td>
<td>94.89</td>
<td>(1.57)</td>
<td>96.14</td>
<td>(2.47)</td>
</tr>
<tr>
<td>9</td>
<td>95.91</td>
<td>(3.27)</td>
<td>94.89</td>
<td>(4.21)</td>
<td>96.37</td>
<td>(4.27)</td>
<td>98.23</td>
<td>(1.61)</td>
</tr>
</tbody>
</table>

4.3. Comparisons under Different Feature Layers

To further reveal the advantages of our DGNMF approach in more detail, this subsection will give the performance evaluation using different layer features on Yale dataset with training number $TN = 3$ and 6. We will consider the three-level decomposition of all deep NMF algorithms. The parameters of all algorithms at the first two levels are set as the same as shown in subsection 4.2. While at the third layer, we let the number of basis images $r^3$ of MNMF, DSNMF, RDNBMF and our DGNMF be $200, 200, 160$ and $20$ respectively. MNMF and DSNMF are the feature matrix-based deep decomposition methods and thus the formulations of their three-layer factorization are the same, namely $X \approx W_1W_2W_3H_3$, where $W_2W_3H_3, W_3H_3$ and $H_3$ will be utilized as their features of layer 1, layer 2 and layer 3 respectively. Especially, MNMF will degrade to single-layer NMF [2] if using the first-layer features. For RDNBMF and DGNMF, they are the basis matrix-based decomposition algorithms and have the same formula of decomposition as $X \approx H_1H_2H_3$, where $H_1$, $H_2H_1$ and $H_3H_2H_1$ will be exploited as their different-level features such as layer 1, layer 2 and layer 3 respectively. The experimental results are listed in Table 3 with $TN = 3$ and Table 4 with $TN = 6$ respectively. We see that the accuracies of four algorithms using high-level features are superior to using the first-layer features under different training numbers. This means the deep NMF algorithms can extract more essential features of the images than the single-layer NMF algorithm. Compared with MNMF, DSNMF, and RDNBMF algorithms, the proposed DGNMF algorithm achieves the best performance under all different conditions, such as different layers and different training numbers.

TABLE 3. Mean Accuracy (%) using Different Layer Features on the Yale database (TN=3)

<table>
<thead>
<tr>
<th>Layer</th>
<th>Layer1</th>
<th>Layer2</th>
<th>Layer3</th>
</tr>
</thead>
<tbody>
<tr>
<td>MNMF</td>
<td>79.34</td>
<td>83.11</td>
<td>82.37</td>
</tr>
<tr>
<td>(std)</td>
<td>(3.02)</td>
<td>(2.35)</td>
<td>(4.94)</td>
</tr>
<tr>
<td>DSNMF</td>
<td>77.32</td>
<td>81.01</td>
<td>79.27</td>
</tr>
<tr>
<td>(std)</td>
<td>(3.91)</td>
<td>(2.27)</td>
<td>(3.32)</td>
</tr>
<tr>
<td>RDNBMF</td>
<td>83.12</td>
<td>85.32</td>
<td>86.31</td>
</tr>
<tr>
<td>(std)</td>
<td>(2.33)</td>
<td>(3.85)</td>
<td>(2.31)</td>
</tr>
<tr>
<td>DGNMF</td>
<td>88.07</td>
<td>93.11</td>
<td>91.45</td>
</tr>
<tr>
<td>(std)</td>
<td>(1.22)</td>
<td>(1.39)</td>
<td>(2.15)</td>
</tr>
</tbody>
</table>
### TABLE 4. Mean Accuracy (%) using Different Layer Features on the Yale database (TN=6)

<table>
<thead>
<tr>
<th>Layer</th>
<th>Layer1</th>
<th>Layer2</th>
<th>Layer3</th>
</tr>
</thead>
<tbody>
<tr>
<td>MNMF</td>
<td>88.53</td>
<td>91.58</td>
<td>89.31</td>
</tr>
<tr>
<td>(std)</td>
<td>(2.37)</td>
<td>(3.95)</td>
<td>(3.32)</td>
</tr>
<tr>
<td>DSNMF</td>
<td>87.71</td>
<td>90.21</td>
<td>88.52</td>
</tr>
<tr>
<td>(std)</td>
<td>(2.92)</td>
<td>(3.06)</td>
<td>(2.41)</td>
</tr>
<tr>
<td>RDNBMF</td>
<td>89.21</td>
<td>94.11</td>
<td>92.43</td>
</tr>
<tr>
<td>(std)</td>
<td>(2.37)</td>
<td>(3.82)</td>
<td>(3.13)</td>
</tr>
<tr>
<td>DGNMF</td>
<td>93.37</td>
<td>97.21</td>
<td>95.32</td>
</tr>
<tr>
<td>(std)</td>
<td>(2.26)</td>
<td>(2.05)</td>
<td>(1.21)</td>
</tr>
</tbody>
</table>

### 5. CONCLUSION

To uncover the underlying hierarchical feature for the image data representation, this paper proposes a novel deep group NMF (DGNMF) approach. The model of DGNMF is established using the knowledge of class label and the locality information of the data. Our DGNMF method automatically makes the data from distinct classes share different basis images and the feature vectors among different classes are mutually orthogonal at the same layer. These good properties are beneficial to improve the performances of the existing deep NMF methods. We develop the DGNMF using gradient descent method and show its convergence via auxiliary function construction strategy. The proposed DGNMF algorithm is finally applied to face recognition. The empirical results exhibit the effectiveness and the outstanding accuracy of the proposed method under different decomposition layers.

### Acknowledgments

This work is partially supported by Natural Science Foundation of Shenzhen (20200815000520001) and the Interdisciplinary Innovation Team of Shenzhen University. We would like to thank Yale University and Amy Research Laboratory for providing the facial image databases.

### References


