Deep Multimodal Subspace Clustering Networks

Mahdi Abavisani, *Student Member, IEEE* and Vishal M. Patel, *Senior Member, IEEE*

***Abstract*—We present convolutional neural network (CNN) based approaches for unsupervised multimodal subspace clustering. The proposed framework consists of three main stages - multimodal encoder, self-expressive layer, and multimodal decoder. The encoder takes multimodal data as input and fuses them to a latent space representation. The self-expressive layer is responsible for enforcing the self-expressiveness property and acquiring an affinity matrix corresponding to the data points. The decoder reconstructs the original input data. The network uses the distance between the decoder's reconstruction and the original input in its training. We investigate early, late and intermediate fusion techniques and propose three different encoders corresponding to them for spatial fusion. The self-expressive layers and multimodal decoders are essentially the same for different spatial fusion-based approaches. In addition to various spatial fusion-based methods, an affinity fusion-based network is also proposed in which the self-expressive layer corresponding to different modalities is enforced to be the same. Extensive experiments on three datasets show that the proposed methods significantly outperform the state-of-the-art multimodal subspace clustering methods.**

***Index Terms*—Deep multimodal subspace clustering, subspace clustering, multimodal learning, multi-view subspace clustering.**

I. Introduction

Many practical applications in image processing, computer vision, and speech processing require one to process very high-dimensional data. However, these data often lie in a low-dimensional subspace. For instance, facial images with variation in illumination [1], handwritten digits [2] and trajectories of a rigidly moving object in a video [3] are examples where the high-dimensional data can be represented by low-dimensional subspaces. Subspace clustering algorithms essentially use this fact to find clusters in different subspaces within a dataset [4]. In other words, in a subspace clustering task, given the data from a union of subspaces, the objective is to find the number of subspaces, their dimensions, the segmentation of the data and a basis for each subspace [4]. This problem has numerous applications including motion segmentation [5], unsupervised image segmentation [6], image representation and compression [7] and face clustering [8].

Various subspace clustering methods have been proposed in the literature [9], [10], [11], [12], [13], [14], [15], [16], [17], [18], [19], [20]. In particular, methods based on sparse and low-rank representation have gained a lot of attraction in recent years [21], [22], [14], [15], [23], [24], [25], [26]. These methods exploit the fact that noiseless data in a union of subspaces are self-expressive, i.e. each data point can be expressed as a sparse linear combination of other data points. The self-expressiveness property was also recently investigated in [16] to develop a deep convolutional neural network (CNN) for subspace clustering. This deep learning-based method was shown to significantly outperform the state-of-the-art subspace clustering methods.

M. Abavisani is with the department of Electrical and Computer Engineering at Rutgers University, Piscataway, NJ USA. email: mahdi.abavisani@rutgers.edu.

Vishal M. Patel is with the department of Electrical and Computer Engineering at Johns Hopkins University, Baltimore, MD USA. email: vpatel36@jhu.edu.

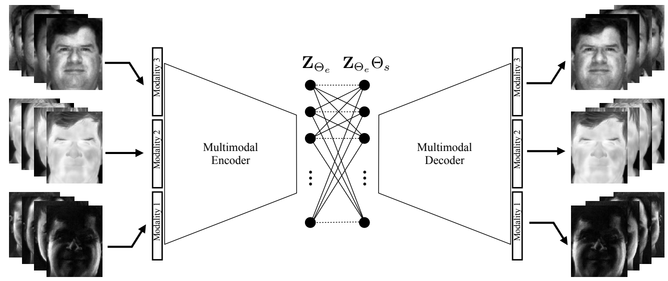


Fig. 1. An overview of the proposed deep multimodal subspace clustering framework. Note that the network consists of three blocks: a multimodal encoder, a self-expressive layer, and a multimodal decoder. The weights in the self-expressive layer, $\mathbf{\Theta}_S$, are used to construct the affinity matrix. We present several models for the multimodal encoder.

In the case where the data consists of multiple modalities or views, multimodal subspace clustering methods can be employed to simultaneously cluster the data in the individual modalities according to their subspaces [27], [28], [29], [30], [31], [32], [33], [34], [35], [36]. Some of the multimodal subspace clustering methods make use of the kernel trick to map the data onto a high-dimensional feature space to achieve better clustering [36].

Motivated by the recent advances in deep subspace clustering [16] as well as multimodal data processing using CNNs [37], [38], [39], [40], [41], [42], [43], [44], [45], [46], in this paper, we propose a different approach to the problem of multimodal subspace clustering. We present a novel CNN-based autoencoder approach in which a fully-connected layer is introduced between the encoder and the decoder which mimics the self-expressiveness property that has been widely used in various subspace clustering algorithms.

Figure 1 gives an overview of the proposed deep multimodal subspace clustering framework. The self-expressive layer is responsible for enforcing the self-expressiveness property and acquiring an affinity matrix corresponding to the data points. The decoder reconstructs the original input data from the latent features. The network uses the distance between the decoder's reconstruction and the original input in its training.

For encoding the multimodal data into a latent space, we investigate three different spatial fusion techniques based on late, early and intermediate fusion. These fusion techniques are motivated by the deep multimodal learning methods in supervised learning tasks [47], [48], that provide the representation of modalities across spatial positions. In addition to the spatial fusion methods, we propose an affinity fusion-based network in which the self-expressive layer corresponding to different modalities is enforced to be the same. For both spatial and the affinity fusion-based methods, we formulate an end-to-end training objective loss.

Key contributions of our work are as follows:

- Deep learning-based multimodal subspace clustering framework is proposed in which the self-expressiveness property is encoded in the latent space by using a fully connected layer.
- Novel encoder network architectures corresponding to late, early and intermediate fusion are proposed for fusing multimodal data.
- An affinity fusion-based network architecture is proposed in which the self-expressive layer is enforced to have the same weights across latent representations of all the modalities.

To the best of our knowledge, this is the first attempt that proposes to use deep learning for multimodal subspace clustering. Furthermore, the proposed method obtains the state-of-the-art results on various multimodal subspace clustering datasets. Code is available at: https://github.com/mahdiabavisani/Deep-multimodal-subspace-clustering-networks.

This paper is organized as follows. Related works on subspace clustering and multimodal learning are presented in Section II. The proposed spatial fusion-based and affinity fusion-based multimodal subspace clustering methods are presented in Section III and IV, respectively. Experimental results are presented in Section V, and finally, Section VI concludes the paper with a brief summary.

II. Related Work

In this section, we review some related works on subspace clustering and multimodal learning.

A. Sparse and Low-rank Representation-based Subspace Clustering

Let $\mathbf{X} = [\mathbf{x}_1, \cdots, \mathbf{x}_N] \in \mathbb{R}^{D\times N}$ be a collection of N signals $\{\mathbf{x}_i \in \mathbb{R}^D\}_{i=1}^N$ drawn from a union of n linear subspaces $\mathcal{S}_1 \cup \mathcal{S}_2 \cup \cdots \cup \mathcal{S}_n$ of dimensions $\{d_\ell\}_{\ell=1}^n$ in $\mathbb{R}^D$. Given $\mathbf{X}$, the task of subspace clustering is to find sub-matrices $\mathbf{X}_\ell \in \mathbb{R}^{D\times N_\ell}$ that lie in $\mathcal{S}_\ell$ with $N_1+N_2+\cdots+N_n = N$. The sparse subspace clustering (SSC) [21] and low-rank representations-based subspace clustering (LRR) [22] algorithms exploit the fact that noiseless data in a union of subspaces are *self-expressive*. In other words, it is assumed that each data point can be represented as a linear combination of other data points. Hence, these algorithms aim to find the sparse or low-rank matrix $\mathbf{C}$ by solving the following optimization problem

$$\min_{\mathbf{C}} \|\mathbf{C}\|_p + \frac{\lambda}{2}\|\mathbf{X} - \mathbf{X}\mathbf{C}\|_F^2, \tag{1}$$

where $\|.\|_p$ is the ℓ_1-norm in the case of SSC [21] and the nuclear norm in the case of LRR [22]. Here, λ is a regularization parameter. In addition, to prevent the trivial solution $\mathbf{C} = \mathbf{I}$, an additional constraint of $\text{diag}(\mathbf{C}) = \mathbf{0}$ is added to the above optimization problem in the case of SSC. Once $\mathbf{C}$ is found, spectral clustering methods [49] are applied on the affinity matrix $\mathbf{W} = |\mathbf{C}| + |\mathbf{C}|^T$ to obtain the segmentation of the data $\mathbf{X}$.

Non-linear versions of the SSC and LRR algorithms have also been proposed in the literature [23], [24].

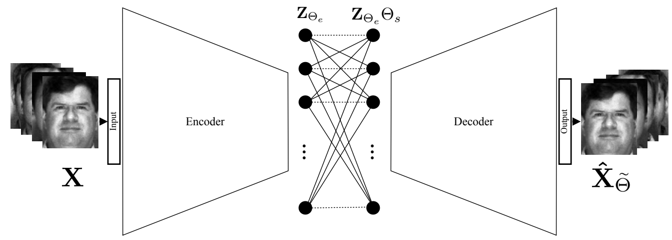


Fig. 2. An overview of the DSC framework proposed in [16] for unimodal subspace clustering.

B. Deep Subspace Clustering

The deep subspace clustering network (DSC) [16] explores the self-expressiveness property by embedding the data into a latent space using an encoder-decoder type network. Figure 2 gives an overview of the DSC method for unimodal subsapce clustering. The method optimizes an objective similar to that of (1) but the matrix $\mathbf{C}$ is approximated using a trainable dense layer embedded within the network. Let us denote the parameters of the self-expressive layer as $\mathbf{\Theta}_s$. Note that these parameters are essentially the elements of $\mathbf{C}$ in (1). The following loss function is used to train the network

$$\min_{\tilde{\mathbf{\Theta}}} \|\mathbf{\Theta}_s\|_p + \frac{\lambda_1}{2}\|\mathbf{Z}_{\mathbf{\Theta}_e} - \mathbf{Z}_{\mathbf{\Theta}_e}\mathbf{\Theta}_s\|_F^2 + \frac{\lambda_2}{2}\|\mathbf{X} - \hat{\mathbf{X}}_{\tilde{\mathbf{\Theta}}}\|, \quad \text{s.t. } \text{diag}(\mathbf{\Theta}_s) = \mathbf{0}, \tag{2}$$

where $\mathbf{Z}_{\mathbf{\Theta}_e}$ denotes the output of the encoder, and $\hat{\mathbf{X}}_{\tilde{\mathbf{\Theta}}}$ is the reconstructed signal at the output of the decoder. Here, the network parameters $\tilde{\mathbf{\Theta}}$ consist of encoder parameters $\mathbf{\Theta}_e$, decoder parameters $\mathbf{\Theta}_d$ and self-expressive layer parameters $\mathbf{\Theta}_s$. Here, λ_1 and λ_2 are two regularization parameters.

C. Multimodal Subspace Clustering

A number of multimodal and multiview subspace clustering approaches have been developed in recent years. Bickel *et al.* introduced an Expectation Maximization (EM) and agglomerative multiview clustering methods in [33]. White *et al.* [32] provided a convex reformulation of multiview subspace learning that as opposed to local formulations enables global learning. Some algorithms use dimensionality reduction methods such as Canonical Correlation Analysis (CCA) to project the multiview data onto a low-dimensional subspace for clustering [28], [34]. Some other multimodal methods are specifically designed for two views and can not be easily generalized to multiple views [50], [35]. Kumar *et al.* [29] proposed a co-regularization method that enforces the clusterings to be aligned in different views. Zhao *et al.* [30] use output of clustering in one view to learn discriminant

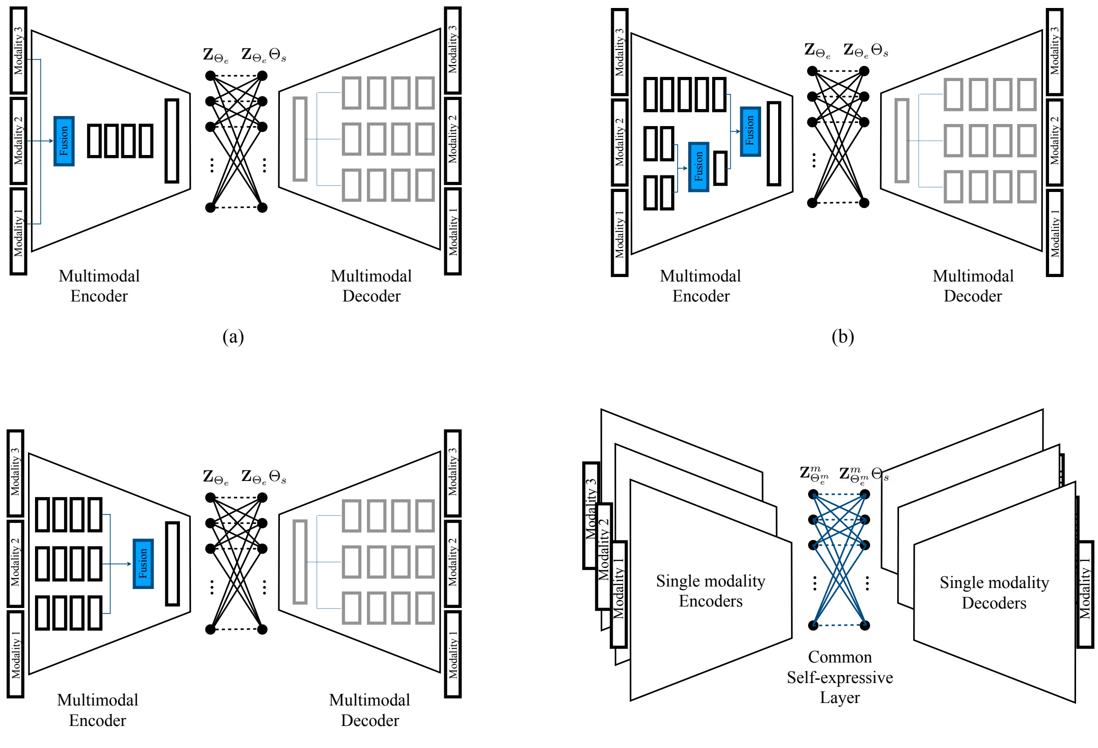


Fig. 3. Different network architectures corresponding to (a) early fusion, (b) intermediate fusion, and (c) late fusion. Note that in all the spatial fusion-based networks (a)-(c), the overall structure for the self-expressive layer and the multimodal decoder remain the same. (d) Network architecture corresponding to affinity fusion. In this case, the encoder and decoder are trained separately for each modality, but are forced to have the same self-expressive layer.

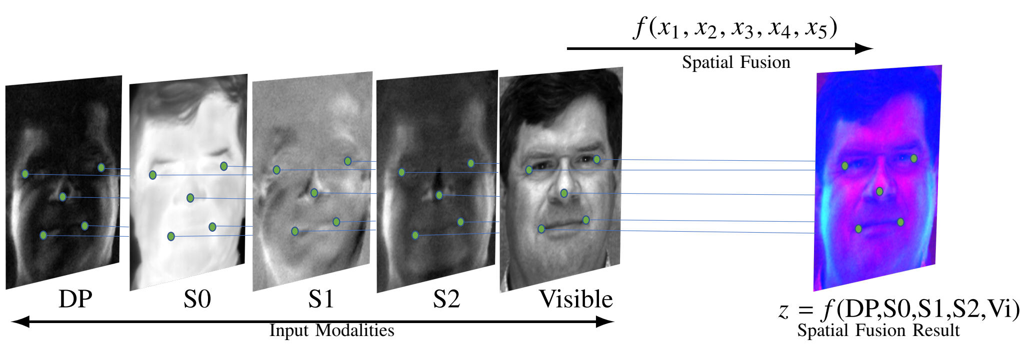


Fig. 4. In spatial fusion methods each location of the fusion is related to the input values at the same location. In this especial case, the facial components (i.e. eyes, nose and mouth) are aligned across all the modalities (i.e. DP, S0, S1, S2, Visible).

subspaces in another view. A multiview subspace clustering method, called Low-rank Tensor constrained Multiview Subspace Clustering (LT-MSC) was recently proposed in [26]. In the LT-MSC method, all the subspace representations are integrated into a low-rank tensor, which captures the high order correlations underlying multiview data. In [51], a diversity-induced multiview subspace clustering was proposed in which the Hilbert Schmidt independence criterion was utilized to explore the complementarity of multiview representations. Recently, [52] proposed a constrained multi-view video face clustering (CMVFC) framework in which pairwise constraints are employed in both sparse subspace representation and spectral clustering procedures for multimodal face clustering. A collaborative image segmentation framework, called Multi-task Low-rank Affinity Pursuit (MLAP) was proposed in [27]. In this method, the sparsity-consistent low-rank affinities from the joint decompositions of multiple feature matrices into pairs of sparse and low-rank matrices are exploited for segmentation.

D. Deep Multimodal Learning

In multimodal learning problems, the idea is to use the complementary information provided by the different modalities to enhance the recognition performance.Supervised deep multimodal learning was first introduced in [37], [38], and has gained a lot of attention in recent years [53], [54], [40].

Keila *et al.* [47] investigated deep multimodal classification of large-scaled datasets. They, compared a number of multimodal fusion methods in terms of accuracy and computational efficiency, and provided analysis regarding the interpretability of multimodal classification models. Feichtenhofer *et al.* [48] proposed a convolutional fusion method for two stream 3D networks. They explored multiple fusion functions within deep architectures and studied the importance of learning the correspondences between spatial and temporal feature maps. Various deep supervised multimodal fusion approaches have also been proposed in the literature for different applications including medical image analysis applications [55], [56] visual recognition [41], [40] and visual question answering [53], [43]. We refer readers to [39] for more detailed survey of various deep supervised multimodal fusion methods.

While most of the deep multimodal approaches have reported improvements in the supervised tasks, to the best of our knowledge, there is no deep multimodal learning method specifically designed for unsupervised subspace clustering.

III. Spatial Fusion-based Deep Multimodal Subspace Clustering

In this section, we present details of the proposed spatial fusion-based networks for unsupervised subspace clustering. Spatial fusion methods find a joint representation that contains complementary information from different modalities. The joint representation has a spatial correspondence to every modality. Figure 4 shows a visual example of spatial fusion where five different modalities (DP, S0, S1, S2, Visible) are combined to produce a fused result z. The spatial fusion methods are especially popular in supervised multimodal learning applications [47], [48]. We investigate applying these fusion techniques to our problem of deep subspace clustering.

An essential component of such methods is the fusion function that merges the information from multiple input representations and returns a fused output. In the case of deep networks, flexibility in the choice of fusion network leads to different models. In what follows, we investigate several network designs and spatial fusion functions for multimodal subspace clustering. Then, we formulate an end-to-end training objective for the proposed networks.

A. Fusion Structures

We build our deep multimodal subspace clustering networks based on the architecture proposed in [16] for unimodal subspace clustering. Our framework consists of three main components: an encoder, a fully connected self-expressive layer, and a decoder. We propose to achieve the spatial fusion using an encoder and the fused representation is then fed to a self-expressive layer which essentially exploits the self-expressiveness property of the joint representation. The joint representation resulting from the output of the self-expressive layer is then fed to a multimodal decoder that reconstructs the different modalities from the joint latent representation.

For the case of M input modalities, the decoder consists of M branches, each reconstructing one of the modalities. The encoders on the other hand, can be designed such that they achieve early, late or intermediate fusion. Early fusion refers to the integration of multimodal data in the stage of feature level before feeding them to the network. Late fusion, on the other hand, involves the integration of multimodal data in the last stage of the network. The flexibility of deep networks also offers the third type of fusion known as the intermediate fusion, where the feature maps from the intermediate layers of a network are combined to achieve better joint representation. Figures 3 (a), (b) and (c) give an overview of deep multimodal subspace clustering networks with different spatial fusion structures. Note that the multimodal decoder's structure remains the same in all three cases. It is worth mentioning that in the case of intermediate fusion, it is a common practice to aggregate the weak or correlated modalities at earlier stages and combine the remaining strong modalities at the in-depth stages [39].

B. Fusion Functions

Assume for a particular data point, x_i, there are M feature maps corresponding to the representation of different modalities. A fusion function $f : \{x^1, x^2. \cdots, x^M\} \to z$ fuses the M feature maps and produces an output z. For simplicity we assume that all the input feature maps have the same dimension of $\mathbb{R}^{H\times W\times d^{in}}$, and the output has the dimension of $\mathbb{R}^{H\times W\times d^{out}}$. In fact, deep network structures offer the design option for having feature maps with the same dimensions. We use $z_{i,j,k}$ and $x^m_{i,j,k}$ to denote the value in the spatial position (i, j, k) in the output and the mth input feature map, respectively. Various fusion functions can be used to combine the input feature maps. Below, we investigate a few.

1) Sum fusion $z = sum(x^1, x^2. \cdots, x^M)$: computes the sum of the feature maps at the same special positions as follows

$$z_{i,j,k} = \sum_{m=1}^{M} x^m_{i,j,k}. \tag{3}$$

2) Maxpooling function $z = max(x^1, x^2. \cdots, x^M)$: returns the maximum value of the corresponding location in the input feature maps as follows

$$z_{i,j,k} = \text{Max}\{x^1_{i,j,k}, x^2_{i,j,k}. \cdots, x^M_{i,j,k}\}. \tag{4}$$

3) Concatenation function $z = cat(x^1, x^2. \cdots, x^M)$: constructs the output by concatenating the input feature maps as follows

$$z = [x^1, x^2. \cdots, x^M], \tag{5}$$

where each input has the dimension $\mathbb{R}^{H\times W\times d_{in}}$ and the output has the dimension $\mathbb{R}^{H\times W\times (d_{in}\times M)}$. Note that these fusion functions are denoted as "Fusion" in blue boxes in Figure 3 (a)-(c).

C. End-to-End Training Objective

Given N paired data samples $\{\mathbf{x}^1_i, \mathbf{x}^2_i, \cdots, \mathbf{x}^M_i\}_{i=1}^N$ from M different modalities, define the corresponding data matrices as $\mathbf{X}^m = [\mathbf{x}^m_1, \mathbf{x}^m_s, \cdots, \mathbf{x}^m_N], \ \ m \in \{1, \cdots, M\}$. Regardless of the network structure and the fusion function of choice, let $\mathbf{\Theta}_{M.e}$ denote the parameters of the multimodal encoder. Similarly, let $\mathbf{\Theta}_s$ be the self-expressive layer parameters and $\mathbf{\Theta}_{M.d}$ be the multimodal decoder parameters. Then the proposed spatial fusion models can be trained end-to-end using the following loss function

$$\min_{\boldsymbol{\Theta}} \|\boldsymbol{\Theta}_s\|_p + \frac{\lambda_1}{2}\|\mathbf{Z}_{\boldsymbol{\Theta}_{M.e}} - \mathbf{Z}_{\boldsymbol{\Theta}_{M.e}}\boldsymbol{\Theta}_s\|_F^2 + \frac{\lambda_2}{2}\sum_{m=1}^{M}\|\mathbf{X}^m - \hat{\mathbf{X}}^m_{\boldsymbol{\Theta}}\| \quad \text{s.t } \mathrm{diag}(\boldsymbol{\Theta}_s) = \mathbf{0}, \quad (6)$$

where $\boldsymbol{\Theta}$ denotes all the training network parameters including $\boldsymbol{\Theta}_{M.e}$, $\boldsymbol{\Theta}_s$ and $\boldsymbol{\Theta}_{M.d}$. The joint representation is denoted by $\mathbf{Z}_{\boldsymbol{\Theta}_{M.e}}$, and $\hat{\mathbf{X}}^m_{\boldsymbol{\Theta}}$ is the reconstruction of $\mathbf{X}^m$. Here, λ_1 and λ_2 are two regularization parameters, and $\|\cdot\|_p$ can be either ℓ_1 or ℓ_2 norm.

Algorithm 1 Spatial and affinity fusion algorithms

procedure DMSC($\{\mathbf{X}^m\}_{m=1}^M$, λ_1, λ_2, `'mode'`)
  if `mode` = Spatial fusion **then**
    Train the networks using the loss (6).
  else if `mode` = Affinity fusion **then**
    Train the networks using the loss (7).
  end if
  Extract $\boldsymbol{\Theta}_s$ from the trained networks.
  Normalize the columns of $\boldsymbol{\Theta}_s$ as $\boldsymbol{\theta}_{si} \leftarrow \frac{\boldsymbol{\theta}_{si}}{\|\boldsymbol{\theta}_{si}\|_\infty}$.
  Form a similarity graph with N nodes and set the weights on the edges by $\mathbf{W} = |\boldsymbol{\Theta}_s| + |\boldsymbol{\Theta}_s^T|$.
  Apply spectral clustering to the similarity graph.
end procedure
Output: Segmented multimodal data.

IV. Affinity Fusion-based Deep Multimodal Subspace Clustering

In this section, we propose a new method for fusing the affinities across the data modalities to achieve better clustering. Spatial fusion methods require the samples from different modalities to be aligned (see Figure 4) to achieve better clustering. In contrast, the proposed affinity fusion approach combines the similarities from the self-expressive layer to obtain a joint representation of the multimodal data. This is done by enforcing the network to have a joint affinity matrix. This avoids the issue of having aligned data or increasing the dimensionality of the fused output (i.e. concatenation). The motivation for enforcing a shared affinity matrix is that similar (dissimilar) data in one modality should be similar (dissimilar) in the other modalities as well. Figure 5 shows an example of the proposed affinity fusion method by forcing the modalities to share the same affinity matrix.

In the DSC framework [16], the affinity matrix is calculated from the self-expressive layer weights as follows

$$\mathbf{W} = |\boldsymbol{\Theta}_s^T| + |\boldsymbol{\Theta}_s^T|,$$

where $\boldsymbol{\Theta}_s$ corresponds to the self-expressive layer weights learned by an end-to-end training strategy [16]. Thus a shared $\boldsymbol{\Theta}_s$ results in a common $\mathbf{W}$ across the modalities. We enforce the modalities to share a common $\boldsymbol{\Theta}_s$ while having different encoders, decoders and the latent representations.

A. Network Structure

For an M modality problem, we propose to stack M parallel DSC networks, where they share a common self-expressive layer. In this model, per each modality one encoder-decoder network is trained. In contrast to the spatial fusion models that only have one joint latent representation, this model results in M distinct latent representations corresponding to M different modalities. The latent representations are connected together by sharing the self-expressive layer. The optimal self-expressive layer should be able to jointly exploit the self-expressiveness property across all the M modalities. Figure 3 (d) gives an overview of the proposed affinity fusion-based network architecture.

B. End-to-End Training

We propose to find the shared self-expressive layer weights by training the network with the following loss

$$\min_{\boldsymbol{\Theta}} \; \|\boldsymbol{\Theta}_s\|_p + \frac{\lambda_1}{2}\sum_{m=1}^{M}\|\mathbf{Z}^m_{\boldsymbol{\Theta}^m_e} - \mathbf{Z}^m_{\boldsymbol{\Theta}^m_e}\boldsymbol{\Theta}_s\|_F^2 + \frac{\lambda_2}{2}\sum_{m=1}^{M}\|\mathbf{X}^m - \hat{\mathbf{X}}^m_{\boldsymbol{\Theta}^m}\| \quad \text{s.t. } \mathrm{diag}(\boldsymbol{\Theta}_s) = \mathbf{0}, \quad (7)$$

where $\boldsymbol{\Theta}_s$ is the common self-expressive layer weighs. Here, λ_1 and λ_2 are regularization parameters. $\mathbf{Z}^m_{\boldsymbol{\Theta}^m_e}$ and $\hat{\mathbf{X}}^m_{\boldsymbol{\Theta}^m}$ are respectively the latent space representation and the reconstructed decoder's output corresponding to $\mathbf{X}^m$. $\boldsymbol{\Theta}^m$ denotes the network parameters corresponding to the mth modality and $\boldsymbol{\Theta}$ indicates to all the trainable parameters. Minimizing (7) encourages the networks to learn the latent representations that share the same affinity matrix.

Algorithm 1 summarizes the proposed spatial fusion and affinity fusion-based subspace clustering methods. Details of different network architectures used in this paper are given in Appendix VI.

V. Experimental Results

We evaluate the proposed deep multimodal subspace clustering methods on several real-world multimodal datasets. The following datasets are used in our experiments.

- Multiview digit clustering using the MNIST [57] and the USPS [58] handwritten digits datasets. Here, we view images from the individual datasets as two views of the same digit. These datasets are considered to be spatially related but not aligned. Since the number of parameters in the self-expressive layer of a deep subspace clustering network scales quadratically with the size of the data, we randomly select 200 samples per digit to keep the networks to a tractable size.
- Heterogeneous face clustering using the ARL Polarimetric face dataset [59]. The ARL dataset contains five spatially well-aligned modalities (Visible, DP, S0, S1, S2).
- Face clustering based on the facial regions using the Extended Yale-B dataset [60]. We extract facial components

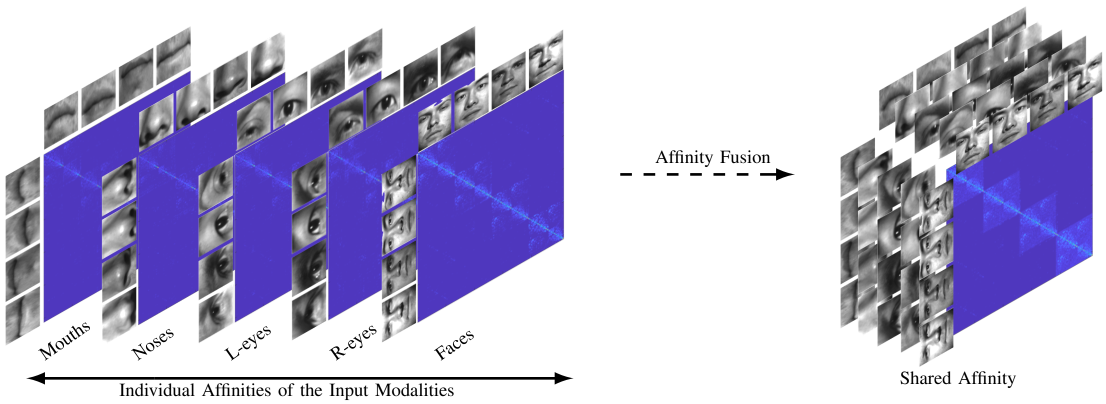


Fig. 5. An example of affinity fusion. Affinities corresponding to different modalities are combined to have only a single shared affinity. This method does not relay on spatial relation across different modalities. Instead, it aggregates the similarities among data points across different modalities and returns a shared affinity.

Experiment	Dataset	# of modalities	# of samples per modality
Digits	MNIST [57], USPS [58]	2	2000
Heterogeneous Faces	ARL [59]	5	2160
Facial components	Extended Yale-B [60]	5	2432

TABLE I
DETAILS OF THE MULTIMODAL DATASETS THAT ARE USED IN THE EXPERIMENTS. NOTE THAT AS OPPOSED TO SUPERVISED METHODS, WE DO NOT SPLIT DATASETS TO TRAINING AND TESTING SETS IN A DEEP SUBSPACE CLUSTERING TASK.

Structure \ Function	Max-pooling	Additive	Concatenation
Early fusion	×	×	Early-concat.
Intermediate fusion	Interm.-mpool.	Interm.-additive	Interm.-concat.
Late fusion	Late-mpool.	Late-additive	Late-concat.

TABLE II
SPATIAL FUSION VARIATIONS THAT ARE USED IN THE EXPERIMENTS.

		DSC[16]	AE+SSC	SSC[21]	LRR[22]
MNIST	ACC	**92.05**	70.1	67.5	67.4
	NMI	**87.07**	80.94	71.64	66.51
	ARI	**84.60**	62.33	57.03	58.33
USPS	ACC	**72.15**	69.9	37.5	44.35
	NMI	74.73	**80.98**	36.61	35.18
	ARI	**65.47**	62.41	28.40	32.11

TABLE III
THE PERFORMANCE OF SINGLE MODALITY SUBSPACE CLUSTERING METHODS ON DIGITS. EXPERIMENTS ARE EVALUATED BY AVERAGE ACC, NMI AND ARI OVER 5 RUNS. WE USE BOLDFACE FOR THE TOP PERFORMER. COLUMNS SPECIFY THE SINGLE MODALITY SUBSPACE CLUSTERING METHOD, AND ROWS SPECIFY THE MODALITY (MNIST OR USPS) AND CRITERIA.

(i.e. eyes, nose, mouth) from the images and view them as soft biometrics and use them along with the entire face for clustering. Here, the modalities do not share any direct spatial correspondence.

Figure 6 (a), (b), and (c) show sample images from the digits, ARL and Extended Yale-B datasets, respectively. Table I gives an overview of their details. Note that as opposed to supervised methods, we do not split datasets into training and testing sets for subspace clustering. Similar to [16], the parameters of the deep subspace clustering networks are trained using the entire dataset.

To investigate ability and limitations of different versions of the proposed fusion methods, we evaluate the affinity fusion method along with a wide range of plausible spatial fusion methods based on different structure designs and fusion functions. For the early fusion structure, we consider the concatenation fusion function[1]. As for the intermediate and late fusion structures, we consider all the three presented fusion functions which results in six distinct models. Table II presents the structural variations we have used for the presented spatial fusion methods and the name we assign to them when reporting their performances. Besides, we compare our methods against the following state-of-the-art multimodal subspace clustering baselines: CMVFC [52], TM-MSC [26], MSSC [36], MLRR [36], KMSSC [36], and KMLRR [36].

[1]Note that applying max-pooling and additive functions in pixel level features might result in information loss.

Also, to explore the contribution of leveraging information from multiple modalities into the performance of subspace clustering task, we report the performance of subspace clustering methods on the single modalities as well. In particular, we report the classical SSC [21] and LRR [22] performances on the individual modalities along with the recently proposed DSC method [16]. Furthermore, we train an encoder-decoder similar to the network in [16] but without the self-expressive layer, and extract the latent space representations. These deep features are then fed to the SSC algorithm for clustering. We call this method “AE+SSC”. This baseline will show the significance of using an end-to-end deep learning method for subspace clustering. In our tables, we use boldface letters to denote the top performing method and specify the corresponding modalities or datasets in the rows, and subspace clustering methods on the columns.

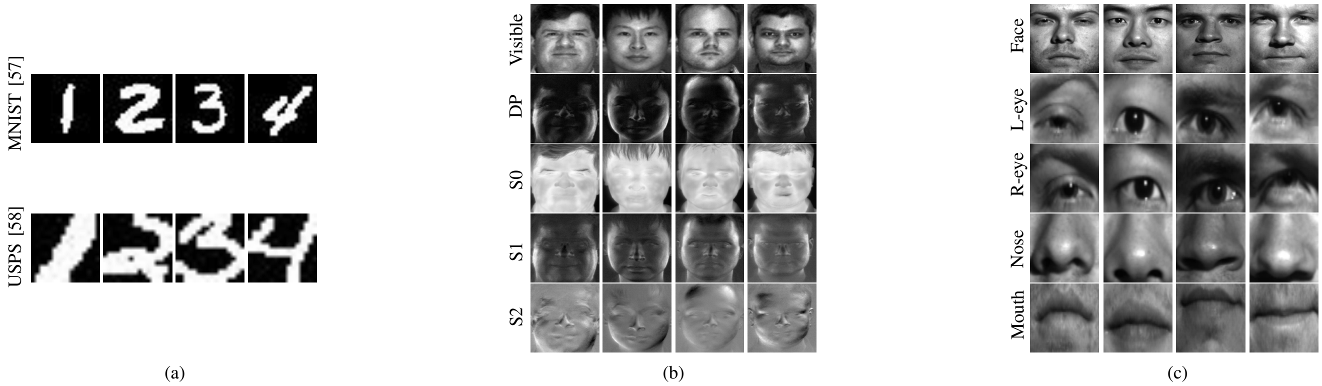


Fig. 6. Sample images from (a) MNIST [57], and USPS [58] digits datasets, (b) ARL polarimetric face dataset [59], and (c) Faces and facial components from the Extended Yale-B dataset [60]. In our experiments, samples from all the modalities are resized to 32×32, and rescaled to have pixel values between 0 and 255.

Structures: We perform all the experiments on different datasets using the same protocol and network architectures to ensure fair and meaningful comparisons (including the networks for the single modality experiments). All the encoders have four convolutional layers, and decoders are stacked three deconvolution layers mimicking the inverse task of the encoder. The network details are given in the Appendix.

For the spatial fusion experiments, in the case of early fusion, we apply the fusion functions on the pixel intensities, and the rest of the network is similar to that of the single modality deep subspace clustering network. Conducted experiments for the intermediate fusion use a prior knowledge on the importance of the modalities. They integrate weak modalities in the second hidden layer, and then, the combination of them in the third layer. Finally, the fusion of all the weak modalities is combined with the strong modality (for example the visible domain in the ARL dataset) in the fourth layer. In the case of late fusion, all the modalities are fused in the fourth layer of the encoder.

As discussed earlier, in the affinity fusion method there exists an encoder-decoder and a latent space per number of available modalities. For example, in the case of the ARL dataset with 5 modalities, we have 5 distinct encoders and decoders connected with a shared self-expressive layer. For each modality in the experiments with the shared affinity, we use similar encoder-decoders as in the case of the DSC network [16] with unimodal experiments.

Training details: We implemented our method in Python-2 with Tensorflow-1.4 [61]. We use the adaptive momentum-based gradient descent method (ADAM) [62] to minimize our loss functions, and apply a learning rate of 10^{-3}.

The input images of all the modalities are resized to 32×32, and rescaled to have pixel values between 0 and 255. In our experiments, the Frobenius norm (i.e. $p = 2$) is used in the loss functions (2), (6) and (7) while training the networks. Similar to [16], for all the methods that have self-expressive layer, we start training on the specified objective functions in each model after a stage of pre-training on the dataset without the self-expressive layer. In particular, for all the proposed deep multimodal subspace clustering methods, and the unimodal DSC networks in the experiments with individual modalities, we pre-train the encoder-decoders for $20k$ epochs with the following objective

$$\min_{\hat{\mathbf{\Theta}}} \sum_{m=1}^{M} \|\mathbf{X}^m - \hat{\mathbf{X}}^m_{\hat{\mathbf{\Theta}}}\|_F^2,$$

where $\hat{\mathbf{\Theta}}$ indicates the union of parameters in the encoder and decoder networks. Note that for the unimodal experiments, $M = 1$.

We use a batch size of 100 for the pretraining stage of all the experiments. However, once we start training the self-expressive layer, the method requires all the data points to be fed as a batch. Thus, in the experiments with digits, ARL faces and Yale-B facial components the batch sizes are 2000, 2160 and 2432, respectively.

We set the regularization parameters as $\lambda_1 = 1$ and $\lambda_2 = 1 \times 10^{\frac{K}{10}-3}$, where K is the number of subjects in the dataset. This experimental rule has been found to be efficient in [16] as well. A sensibility analysis over the range $[10^{-4}, 10^4]$ in Section V-E, shows that if λ_1 and λ_2 are kept around the same scale as our selections, the performance of the proposed method is not much sensitive to these parameters for a set of wide ranges.

Evaluation metrics: We compare the performance of different methods using the clustering accuracy rate (ACC), normalized mutual information (NMI) [63], and Adjusted Rand Index (ARI) [64] metrics.

In external validation of clustering methods where ground truth labels are available, a correct clustering is usually referred as assigning objects belonging to the same category in the ground truth to the same cluster, and objects belonging to different categories to different clusters. With that, ACC is defined as the number of data points correctly clustered divided by the total number of data points. The ARI metric, in addition to penalizing the misclustered data points, penalizes putting two objects with the same label in different clusters, and is adjusted such that a random clustering will score close to 0. The NMI captures the mutual information between the correct labels and the predicted labels, and is normalized between the range [0,1].

A. Handwritten Digits

In the first set of experiments, we use the 10 classes (i.e. digits) from the MNIST and the USPS datasets. Figure 6 (a) shows example images from these datasets. For the experiments with digits, we randomly sample 200 images per class from their training sets to reduce the computations and adjust the imbalance in the tests.

We randomly bundle the same class samples across the two datasets and assume they present two modalities (views) of a digit. One can see from Figure 6 (a), that the needed receptive field for recognizing the digits in the MNIST and the USPS datasets is relatively large. Based on this logic, in the experiments with digits, we use large kernels in the encoders. The detailed network settings for these experiments are described in the Appendix. Note that some structures including the late fusion methods in Table II and the affinity fusion method have more than one branches in some of their layers.

Table III shows the performance of deep subspace clustering per individual digits. This table reveals that the MNIST dataset is easier than the USPS dataset for the subspace clustering task. This observance coincides with the performance of other methods reported in [65].

Note that while the DSC method in Table III shows the-state-of-the-art performance on both datasets, a successful multimodal method should enhance the performance by leveraging the information across the two modalities. Table IV compares the performance of the multimodal methods in terms of accuracy, NMI and ARI metrics. We observe that most of the multimodal methods can successfully integrate the complementary information of the datasets in the subspace clustering task and provide a better performance in comparison to their unimodal counterpart. However, the proposed deep multimodal subspace clustering methods perform significantly better than the classical multimodal subspace clustering methods. In particular, the affinity fusion and late-addition methods can segment the digits with an accuracy of 95.15%, and NMI and ARI metric of above 90%.

B. ARL Heterogeneous Face Dataset

To test our methods on clustering datasets with a large number of subjects, we use the ARL dataset [59] which consists of facial images from 60 unique individuals in different spectrums and from different distances. This dataset has facial images in the visible domain as well as four different polarimetric thermal domains. Each subject has several well-aligned facial images per each modality. Sample images from this dataset are shown in Figure 6 (b).

Table V compares the performance of subspace clustering methods on individual modalities in the ARL dataset. As expected, the visible modality shows better performance among the different spectrums. As the samples are well-aligned in this dataset, we see that most of the subspace clustering methods work well across all the modalities. In particular, the LRR method which takes the advantage of aligned data points, provides comparable results to the DSC method.

		DSC[16]	AE+SSC	SSC[21]	LRR[22]
Visible	ACC	**92.54**	89.87	81.86	91.07
	NMI	**97.03**	96.25	94.56	97.16
	ARI	**92.54**	88.08	72.32	89.94
DP	ACC	**91.81**	89.08	63.2	89.4
	NMI	**97.60**	97.17	83.59	95.71
	ARI	**91.69**	87.48	47.98	85.47
S0	ACC	**62.64**	55.38	21.58	57.23
	NMI	**84.20**	77.62	47.83	80.44
	ARI	**49.23**	41.60	11.63	36.56
S1	ACC	**91.72**	86.21	54.68	86.12
	NMI	**97.09**	96.55	78.60	95.13
	ARI	**89.55**	86.16	42.69	85.62
S2	ACC	**89.68**	89.26	57.92	85.88
	NMI	**97.63**	97.38	82.77	94.73
	ARI	**89.34**	88.05	43.38	84.05

TABLE V
THE PERFORMANCE OF SINGLE MODALITY SUBSPACE CLUSTERING METHODS ON ARL DATASET. EXPERIMENTS ARE EVALUATED BY AVERAGE ACC, NMI AND ARI OVER 5 RUNS. WE USE BOLDFACE FOR THE TOP PERFORMER. COLUMNS SPECIFY THE SINGLE MODALITY SUBSPACE CLUSTERING METHOD, AND ROWS SPECIFY THE MODALITIES AND CRITERIA.

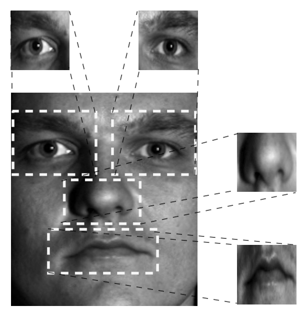


Fig. 7. Facial components are extracted by applying a fixed mask on the faces in the Extended Yale-B dataset [60].

Since the ARL dataset has multiple modalities, beside the early and late fusion structures, we also use an intermediate structure when designing the multimodal encoders. Hence, in this experiment, we add the following intermediate spatial fusion structure to the multimodal methods. Assuming the visible domain is the main modality, we integrate $S0$, $S1$ and $S2$ modalities in the second layer and combine their fused output with the DP samples in the third layer. Finally, we fuse the result with the visible domain at the last layer of the encoders.

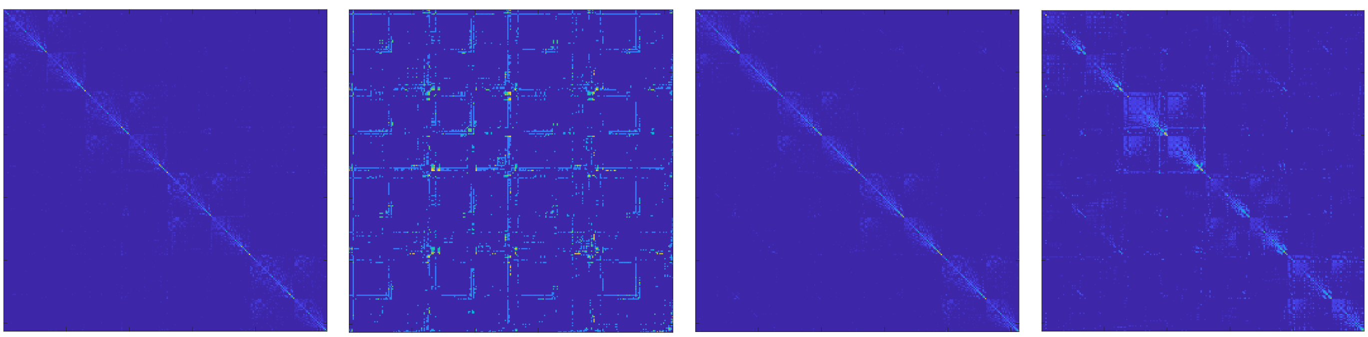
The performances of deep multimodal subspace clustering methods are compared in Table IV. We observe that most of the methods are able to leverage the complementary information of the different spectrums and provide a more accurate clustering in comparison to the unimodal performances. In particular, the affinity fusion method has the best performance, and late-concat and early-concat methods provide comparable results. This experiment clearly shows that our proposed methods can perform well even with a large number of subjects in the dataset.

		CMVFC[52]	TM-MSC[26]	MSSC[36]	MLRR[36]	KMSSC[36]	KMLRR[36]	Early-concat.
Digits	ACC	47.6	80.65	81.65	80.6	84.4	86.85	92.2
	NMI	73.56	83.44	85.33	84.13	89.45	80.34	88.53
	ARI	38.12	75.67	77.36	76.53	79.61	82.76	84.60
ARL	ACC	96.58	96.64	97.78	97.5	97.97	97.74	98.24
	NMI	98.39	98.35	99.58	99.57	99.51	99.58	99.27
	ARI	94.85	95.85	96.40	95.79	96.09	95.88	97.21
Extended Yale-B	ACC	66.84	63.12	80.3	67.62	87.65	82.45	65.55
	NMI	72.03	67.06	82.78	73.36	81.50	85.43	78.82
	ARI	40.00	38.37	50.18	40.85	63.83	59.71	41.95

		Interm.-concat.	Interm-addition	Interm.-mpool.	Late-concat.	Late-addition	Late-mpool	Affinity fusion
Digits	ACC	N/A	N/A	N/A	91.15	**95.15**	91.45	**95.15**
	NMI	N/A	N/A	N/A	84.28	91.35	89.32	**92.09**
	ARI	N/A	N/A	N/A	85.46	89.72	87.74	**90.22**
ARL	ACC	97.79	96.21	94.99	98.22	96.68	95.77	**98.34**
	NMI	**99.59**	98.95	98.19	99.31	99.23	98.92	99.36
	ARI	95.85	94.64	92.93	97.02	96.24	94.77	**97.51**
Extended Yale-B	ACC	94.88	97.65	7.76	92.45	67.41	7.06	**99.22**
	NMI	93.90	96.88	9.31	92.53	66.95	6.39	**98.89**
	ARI	88.19	94.96	0.73	82.91	33.37	00.48	**98.38**

• N/A indicates that the corresponding method is not applicable to this experiment.

TABLE IV
THE PERFORMANCE OF MULTIMODAL SUBSPACE CLUSTERING METHODS. EACH EXPERIMENT IS EVALUATED BY AVERAGE ACC, NMI AND ARI OVER 5 RUNS. WE USE BOLDFACE FOR THE TOP PERFORMER. COLUMNS OF THIS TABLE SHOW THE MULTIMODAL SUBSPACE CLUSTERING METHOD, AND THE ROWS LIST DATASETS AND CLUSTERING METRICS.



(a) (b) (c) (d)

Fig. 8. Visualization of the affinity matrices for first four subjects in the Extended Yale-B dataset calculated from the self-expressive layer weight matrices in (a) unimodal clustering on faces using DSC. (b) The *late-mpool* method. (c) The *late-concat* method. (d) The *affinity fusion* method. Note that (b) shows a failure case of the spatial fusion methods.

C. Facial Components

The Extended Yale-B dataset [60] consists of 64 frontal images of 38 individuals under varying illumination conditions. This dataset is popular in subspace clustering studies [16], [22], [21]. We crop the facial components (i.e. eyes, nose and mouth), and view them as weak modalities. In the biometrics literature, they are viewed as soft biometrics [66]. To crop the facial components, we apply a fixed face mask as shown in Figure 7 on all the facial images. The extracted facial regions are resized to 32 × 32 images. This experiment is especially important as the modalities do not share the spatial correspondence. For example, spatial locations in the mouth modality cannot be projected on the spatial positions in the nose modality. Sample images from this dataset are shown in Figure 6 (c). The setting in this experiment can examine the proposed methods under the condition of spatially unrelated modalities.

The performance of subspace clustering methods on the individual facial components is summarized in Table VI. We observe that the nose and the mouth modalities fail to provide good clustering results. On the other hand, DSC and AE+SSC perform well on the eye and the entire face modalities.

Since the mouth, nose, and eyes are considered as weak modalities, in the design of the intermediate spatial fusion we combine the two eyes, and the mouth and the nose separately in the second layer of the encoders, and fuse the result of their combinations in the third layer. Finally, we fuse the combined features with the face features in the fourth layer.

The performance of various multimodal subspace clustering methods are tabulated in Table IV. It is worth highlighting several interesting observations from the results. As can be seen, the *late-mpool* and *interm-mpool* methods fail to segment the data points. That is because this fusion function at each spatial position returns the maximum of the activation values at the same spatial position between its input feature maps. Since the modalities do not share any spatial correspondence in this experiment, this function does not provide good performance. In addition, even though additive and concatenate

		DSC[16]	AE+SSC	SSC[21]	LRR[22]
Face	ACC	**96.82**	72.93	72.78	63.34
	NMI	**94.82**	79.10	79.17	70.08
	ARI	**91.31**	43.94	42.90	37.38
Right-eye	ACC	**87.62**	83.34	66.84	65.35
	NMI	**89.19**	86.99	73.62	69.33
	ARI	**75.05**	61.90	39.66	38.37
Left-eye	ACC	**80.94**	72.24	63.02	63.08
	NMI	**79.58**	76.48	69.08	70.13
	ARI	**50.17**	42.94	33.12	34.07
Nose	ACC	**67.53**	51.61	41.51	39.9
	NMI	**75.23**	61.64	50.78	48.73
	ARI	**40.82**	22.96	16.67	15.13
Mouth	ACC	**76.86**	67.42	56.07	62.92
	NMI	**76.42**	72.91	64.11	67.28
	ARI	**43.90**	40.52	25.71	33.02

TABLE VI
THE PERFORMANCE OF SINGLE MODALITY SUBSPACE CLUSTERING METHODS ON EXTENDED YALE-B DATASET. EXPERIMENTS ARE EVALUATED BY AVERAGE ACC, NMI AND ARI OVER 5 RUNS. WE USE BOLDFACE FOR THE TOP PERFORMER. COLUMNS SPECIFY THE SINGLE MODALITY SUBSPACE CLUSTERING METHOD, AND ROWS SPECIFY THE FACIA COMPONENTS AND CRITERIA.

fusion functions have provided good results in some cases, because of a similar reason their performances are highly related to the structure choices. For example, the additive function provides better performance with the intermediate fusion structure, while the concatenation works better with the late fusion structure choice. However, the *affinity fusion* provides the state-of-the-art clustering performance of above 99% accuracy, the NMI of 98.89% and ARI metric of 98.38%. This is mainly due to the fact that this method does not rely on the spatial correspondence among the modalities.

Figure 8 compares the affinity matrices of the first four subjects in the Extended Yale-B datasets. The affinity matrices are calculated from the self-expressive layer weights of their corresponding trained networks. The depicted affinity matrices in these figures are the result of a permutation being applied on the matrix so that data points of the same clusters are alongside each other. With this arrangement, a perfect affinity matrix should be block diagonal.

Figure 8 (a) shows the affinity matrix corresponding to the DSC method for clustering faces. Figure 8 (b) shows this matrix for the multimodal subspace clustering with the *late-mpool* method. Note that this method fails to cluster the data, and as can be seen, its affinity matrix is not block-diagonal. Figure 8 (c) and Figure 8 (d) show the affinity matrices of the *late-concat* and *affinity fusion* methods, respectively. We observe that both methods provide a solid block diagonal affinity matrices.

D. Convergence study

To empirically show the convergence of our proposed method, in Figure 9, we show the objective function of the *affinity fusion* method and its clustering metrics vs iteration plot for solving (7). The reported values in Figure 9 are normalized between zero and one. As can be seen from the figure, our algorithm converges in a few iterations.

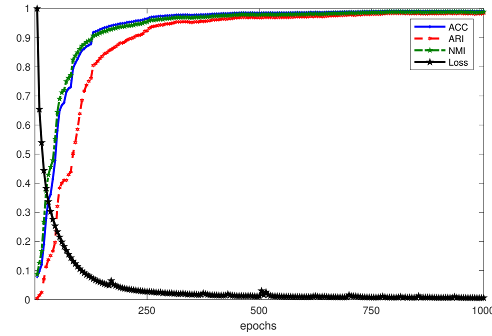


Fig. 9. The *affinity fusion* method's loss function and the clustering metrics over different training epochs in the Yale-B facial components experiment. The reported values in this figure are normalized between zero and one. This figure shows the convergence of our objective function.

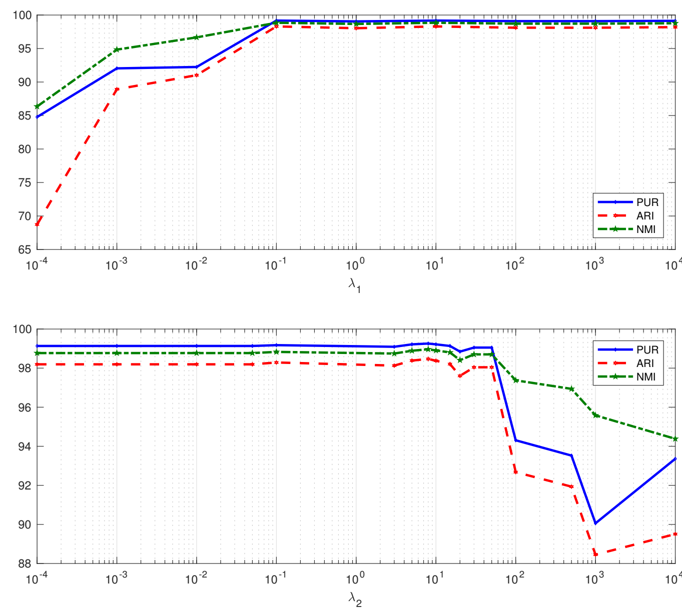


Fig. 10. The *affinity fusion* method's performance through different parameter selections for λ_1 and λ_2.

E. Regularization parameters

In this section, we analyze the sensibility of the proposed method to the regularization parameters λ_1 and λ_2 in the loss function (7). Figure 10 shows the influence of these regularization parameters on the performance of the *affinity fusion* method on the Extended Yale-B dataset.

In Figure 10 (a), we fix $\lambda_2 = 1$ and report the metrics with various λ_1s over the range of $[10^{-4}, 10^4]$. Similarly, in Figure 10 (b), we fix $\lambda_1 = 1$ and this time change λ_2 in the similar range to analyze the influence of λ_2 on the performance of the method. As can be seen from the figure, in a wide range of values, the final performance of the method is not sensitive to the choice of parameters. The experimental setting suggested in [16] also performed well in all the experiments.

F. Performance with respect to different norms on the self-expressive layer

In this section, we compare the performance of the proposed *affinity fusion* method by changing the p-norm on the self-expressive layer in the optimization problem (7). Table VII

Metric \ $\|\cdot\|_p$	$p < 0.3$	$p = 0.3$	$p = 1$	$p = 1.5$	$p = 2$
PUR	×	09.32	99.13	99.17	**99.22**
NMI	×	18.64	98.78	98.84	**98.89**
ARI	×	02.38	98.20	98.29	**98.38**

TABLE VII
ANALYSIS OF DIFFERENT REGULARIZATION NORMS ON THE SELF-EXPRESSIVE LAYER. OUR EXPERIMENTS WITH $p < 0$ DID NOT CONVERGED. THE RESULTS ARE 5-FOLD AVERAGE. WE USE BOLDFACE FOR THE TOP PERFORMER.

reports the clustering metrics for the experiments with $p = 0.3$, $p = 1$, $p = 1.5$ and $p = 2$. As can be seen from this table, while experiments with $p = 1$, $p = 1.5$ and $p = 2$ have comparable performances, applying the p-norm with $p = 0.3$ does not provide sufficient result. It is worth mentioning that in our experiments with different norms with $0.3 < p < 1$ the method showed instability, and for $p < 0.3$ the minimization of (7) did not converge. The reason is that the norms with $p < 1$ are non-convex, and one might need additional regularizations to keep the optimization tractable.

VI. Conclusion

We presented novel deep multimodal subspace clustering networks for clustering multimodal data. In particular, we presented two fusion techniques of spatial fusion and affinity fusion. We observed that spatial fusion methods in a deep multimodal subspace clustering task relay on spatial correspondences among the modalities. On the other hand, the proposed affinity fusion that finds a shared affinity across all the modalities provides the state-of-the-art results in all the conducted experiments. This method clusters the images in the Extended Yale-B dataset with an accuracy of 99.22%, normalized mutual information of 98.89% and adjusted rand index of 98.38%.

Acknowledgment

This work was supported by US Office of Naval Research (ONR) Grant YIP N00014-16-1-3134.

Appendix: Network Architectures

In this section, we provide the details of the network architecture used in the experiments. Note that all the plugged in convolutional layers use *relu* as well.

A. Different networks corresponding to digits experiments

TABLE VIII
EARLY-FUSION NETWORKS IN THE DIGITS EXPERIMENTS.

	Layer	Input	output	Kernel	(stride, pad)
Feature Fusion	Fusion 1	Image 1 Image 2	Fusion 1	-	-
Convolutional layers	Conv 1	Fusion 1	Conv 1	$1 \times 7 \times 7 \times 7$	(2,1)
	Conv 2	Conv 1	Conv 2	$1 \times 5 \times 5 \times 10$	(2,1)
	Conv 3	Conv 2	Conv 3	$1 \times 3 \times 3 \times 15$	(1,0)
	Conv 4	Conv 3	Latent	$1 \times 1 \times 1 \times 15$	(1,0)
Self-expressiveness	Θ_s	Latent	L-recon	4000000 Parameters	-
Multimodal Decoder	Decoder layers	L-recon	Recon 1 Recon 2	Details in Table XI	

TABLE IX
LATE-FUSION NETWORKS IN THE DIGITS EXPERIMENTS.

	Layer	Input	output	Kernel	(stride, pad)
Branch 1	B1/Conv 1	Image 1	B1/Conv 1	$1 \times 7 \times 7 \times 7$	(2,1)
	B1/Conv 2	B1/Conv 1	B1/Conv 2	$1 \times 5 \times 5 \times 10$	(2,1)
	B1/Conv 3	B1/Conv 2	B1/Conv 3	$1 \times 3 \times 3 \times 15$	(1,0)
	B1/Conv 4	B1/Conv 3	B1/out	$1 \times 1 \times 1 \times 15$	(1,0)
Branch 2	B2/Conv 1	Image 2	B2/Conv 1	$1 \times 7 \times 7 \times 7$	(2,1)
	B2/Conv 2	B2/Conv 1	B2/Conv 2	$1 \times 5 \times 5 \times 10$	(2,1)
	B2/Conv 3	B2/Conv 2	B2/Conv 3	$1 \times 3 \times 3 \times 15$	(1,0)
	B2/Conv 4	B2/Conv 3	B2/out	$1 \times 1 \times 1 \times 15$	(1,0)
Feature Fusion	Fusion 1	B1/out B2/out	Latent	-	-
Self-expressiveness	Θ_s	Latent	L-recon	4000000 Parameters	-
Multimodal Decoder	Decoder layers	L-recon	Recon 1 Recon 2	Details in Table XI	

TABLE X
AFFINITY FUSION NETWORKS IN THE DIGITS EXPERIMENTS.

	Layer	Input	output	Kernel	(stride, pad)
Encoder 1	B1/Conv 1	Image 1	B1/Conv 1	$1 \times 7 \times 7 \times 7$	(2,1)
	B1/Conv 2	B1/Conv 1	B1/Conv 2	$1 \times 5 \times 5 \times 10$	(2,1)
	B1/Conv 3	B1/Conv 2	B1/Conv 3	$1 \times 3 \times 3 \times 30$	(1,0)
	B1/Conv 4	B1/Conv 3	Latent 1	$1 \times 3 \times 3 \times 30$	(1,0)
Encoder 2	B2/Conv 1	Image 2	B2/Conv 1	$1 \times 7 \times 7 \times 7$	(2,1)
	B2/Conv 2	B2/Conv 1	B2/Conv 2	$1 \times 5 \times 5 \times 10$	(2,1)
	B2/Conv 3	B2/Conv 2	B2/Conv 3	$1 \times 3 \times 3 \times 15$	(1,0)
	B2/Conv 4	B2/Conv 3	Latent 2	$1 \times 3 \times 3 \times 15$	(1,0)
Self-expressiveness layer	Common Θ_s	Latent 1 Latent 2	L-recon 1 L-recon 2	4000000 Parameters	-
Decoder 1	D1/deconv 1	L-recon 1	D1/deconv 1	$1 \times 3 \times 3 \times 15$	(1,0)
	D1/deconv 2	D1/deconv 1	D1/deconv 2	$1 \times 5 \times 5 \times 10$	(2,1)
	D1/deconv 3	D1/deconv 2	Recon 1	$1 \times 7 \times 7 \times 7$	(2,1)
Decoder 2	D2/deconv 2	L-recon 2	D2/deconv 1	$1 \times 3 \times 3 \times 15$	(1,0)
	D2/deconv 2	D2/deconv 1	D2/deconv 2	$1 \times 5 \times 5 \times 10$	(2,1)
	D2/deconv 3	D2/deconv 2	Recon 2	$1 \times 7 \times 7 \times 7$	(2,1)

TABLE XI
MULTIMODAL DECODER IN THE DIGITS EXPERIMENTS.

	Layer	Input	output	Kernel	(stride, pad)
Decoder 1	D1/deconv 1	L-recon	D1/deconv 1	$1 \times 3 \times 3 \times 15$	(1,0)
	D1/deconv 2	D1/deconv 1	D1/deconv 2	$1 \times 5 \times 5 \times 10$	(2,1)
	D1/deconv 3	D1/deconv 2	Recon 1	$1 \times 7 \times 7 \times 7$	(2,1)
Decoder 2	D2/deconv 2	L-recon	D2/deconv 1	$1 \times 3 \times 3 \times 15$	(1,0)
	D2/deconv 2	D2/deconv 1	D2/deconv 2	$1 \times 5 \times 5 \times 10$	(2,1)
	D2/deconv 3	D2/deconv 2	Recon 2	$1 \times 7 \times 7 \times 7$	(2,1)

B. Different networks corresponding to ARL experiments

TABLE XII
EARLY-FUSION NETWORKS IN THE ARL EXPERIMENTS.

	Layer	Input	output	Kernel	(stride, pad)
Feature Fusion	Fusion 1	Image 1 Image 2 Image 3 Image 4 Image 5	Fusion 1	-	-
Convolutional layers	Conv 1 Conv 2 Conv 3 Conv 4	Fusion 1 Conv 1 Conv 2 Conv 3	Conv 1 Conv 2 Conv 3 Latent	$1 \times 3 \times 3 \times 5$ $1 \times 1 \times 1 \times 7$ $1 \times 1 \times 1 \times 15$ $1 \times 1 \times 1 \times 15$	(2,1) (2,1) (1,0) (1,0)
Self-expressiveness	Θ_s	Latent	L-recon	4665600 Parameters	-
Multimodal Decoder	Decoder layers	L-recon	Recon 1 Recon 2 Recon 3 Recon 4 Recon 5	Details in Table XVI	

TABLE XIII
LATE-FUSION NETWORKS IN THE ARL EXPERIMENTS.

	Layer	Input	output	Kernel	(stride, pad)
Branch 1	B1/Conv 1 B1/Conv 2 B1/Conv 3 B1/Conv 4	Image 1 B1/Conv 1 B1/Conv 2 B1/Conv 3	B1/Conv 1 B1/Conv 2 B1/Conv 3 B1/out	$1 \times 3 \times 3 \times 5$ $1 \times 1 \times 1 \times 7$ $1 \times 1 \times 1 \times 15$ $1 \times 1 \times 1 \times 15$	(2,1) (2,1) (1,0) (1,0)
Branch 2	B2/Conv 1 B2/Conv 2 B2/Conv 3 B2/Conv 4	Image 2 B2/Conv 1 B2/Conv 2 B2/Conv 3	B2/Conv 1 B2/Conv 2 B2/Conv 3 B2/out	$1 \times 3 \times 3 \times 5$ $1 \times 1 \times 1 \times 7$ $1 \times 1 \times 1 \times 15$ $1 \times 1 \times 1 \times 15$	(2,1) (2,1) (1,0) (1,0)
Branch 3	B3/Conv 1 B3/Conv 2 B3/Conv 3 B3/Conv 4	Image 3 B3/Conv 1 B3/Conv 2 B3/Conv 3	B3/Conv 1 B3/Conv 2 B3/Conv 3 B3/out	$1 \times 3 \times 3 \times 5$ $1 \times 1 \times 1 \times 7$ $1 \times 1 \times 1 \times 15$ $1 \times 1 \times 1 \times 15$	(2,1) (2,1) (1,0) (1,0)
Branch 4	B4/Conv 1 B4/Conv 2 B4/Conv 3 B4/Conv 4	Image 4 B4/Conv 1 B4/Conv 2 B4/Conv 3	B4/Conv 1 B4/Conv 2 B4/Conv 3 B4/out	$1 \times 3 \times 3 \times 5$ $1 \times 1 \times 1 \times 7$ $1 \times 1 \times 1 \times 15$ $1 \times 1 \times 1 \times 15$	(2,1) (2,1) (1,0) (1,0)
Branch 5	B5/Conv 1 B5/Conv 2 B5/Conv 3 B5/Conv 4	Image 5 B5/Conv 1 B5/Conv 2 B5/Conv 3	B5/Conv 1 B5/Conv 2 B5/Conv 3 B5/out	$1 \times 3 \times 3 \times 5$ $1 \times 1 \times 1 \times 7$ $1 \times 1 \times 1 \times 15$ $1 \times 1 \times 1 \times 15$	(2,1) (2,1) (1,0) (1,0)
Feature Fusion	Fusion 1	B1/out B2/out B3/out B4/out B5/out	Latent	-	-
Self-expressiveness	Θ_s	Latent	L-recon	4665600 Parameters	-
Multimodal Decoder	Decoder layers	L-recon	Recon 1 Recon 2 Recon 3 Recon 4 Recon 5	Details in Table XVI	

TABLE XIV
INTERMEDIATE SPATIAL FUSION NETWORKS IN THE ARL EXPERIMENTS.

	Layer	Input	output	Kernel	(stride, pad)
Layer 1	B1/Conv 1 B2/Conv 1 B3/Conv 1 B4/Conv 1 B5/Conv 1	Image 1 Image 2 Image 3 Image 4 Image 5	B1/Conv 1 B2/Conv 1 B3/Conv 1 B4/Conv 1 B5/Conv 1	$1 \times 3 \times 3 \times 5$ $1 \times 3 \times 3 \times 5$ $1 \times 3 \times 3 \times 5$ $1 \times 3 \times 3 \times 5$ $1 \times 3 \times 3 \times 5$	(2,1) (2,1) (2,1) (2,1) (2,1)
Feature Fusion	B345/Fusion	B3/Conv 1 B4/Conv 1 B5/Conv 1	B345/Fusion	-	-
Layer 2	B1/Conv 2 B2/Conv 2 B345/Conv 2	B1/Conv 1 B2/Conv 1 B345/Fusion	B1/Conv 2 B2/Conv 2 B345/Conv 2	$1 \times 1 \times 1 \times 7$ $1 \times 1 \times 1 \times 7$ $1 \times 1 \times 1 \times 7$	(2,1) (2,1) (2,1)
Feature Fusion	B2345/Fusion	B345/Conv 2 B2/Conv 2	B2345/Fusion	-	-
Layer 3	B1/Conv 3 B2345/Conv 3	B1/Conv 2 B2345/Fusion	B1/Conv 3 B2345/Conv 3	$1 \times 1 \times 1 \times 15$ $1 \times 1 \times 1 \times 15$	(1,0) (1,0)
Feature Fusion	Ball/Fusion	B1/Conv 3 B2345/Conv 3	Ball/Fusion	-	-
Layer 4	Ball/Conv 4	Ball/Conv 3	Latent	$1 \times 1 \times 1 \times 15$	(1,0)
Self-expressiveness	Θ_s	Latent	L-recon	4665600 Parameters	-
Multimodal Decoder	Decoder layers	L-recon	Recon 1 Recon 2 Recon 3 Recon 4 Recon 5	Details in Table XVI	

TABLE XV
AFFINITY FUSION NETWORKS IN THE ARL EXPERIMENTS.

	Layer	Input	output	Kernel	(stride, pad)
Encoder 1	B1/Conv 1 B1/Conv 2 B1/Conv 3 B1/Conv 4	Image 1 B1/Conv 1 B1/Conv 2 B1/Conv 3	B1/Conv 1 B1/Conv 2 B1/Conv 3 Latent 1	$1 \times 3 \times 3 \times 5$ $1 \times 1 \times 1 \times 7$ $1 \times 1 \times 1 \times 15$ $1 \times 1 \times 1 \times 15$	(2,1) (2,1) (1,0) (1,0)
Encoder 2	B2/Conv 1 B2/Conv 2 B2/Conv 3 B2/Conv 4	Image 2 B2/Conv 1 B2/Conv 2 B2/Conv 3	B2/Conv 1 B2/Conv 2 B2/Conv 3 Latent 2	$1 \times 3 \times 3 \times 5$ $1 \times 1 \times 1 \times 7$ $1 \times 1 \times 1 \times 15$ $1 \times 1 \times 1 \times 15$	(2,1) (2,1) (1,0) (1,0)
Encoder 3	B3/Conv 1 B3/Conv 2 B3/Conv 3 B3/Conv 4	Image 3 B3/Conv 1 B3/Conv 2 B3/Conv 3	B3/Conv 1 B3/Conv 2 B3/Conv 3 Latent 3	$1 \times 3 \times 3 \times 5$ $1 \times 1 \times 1 \times 7$ $1 \times 1 \times 1 \times 15$ $1 \times 1 \times 1 \times 15$	(2,1) (2,1) (1,0) (1,0)
Encoder 4	B4/Conv 1 B4/Conv 2 B4/Conv 3 B4/Conv 4	Image 4 B4/Conv 1 B4/Conv 2 B4/Conv 3	B4/Conv 1 B4/Conv 2 B4/Conv 3 Latent 4	$1 \times 3 \times 3 \times 5$ $1 \times 1 \times 1 \times 7$ $1 \times 1 \times 1 \times 15$ $1 \times 1 \times 1 \times 15$	(2,1) (2,1) (1,0) (1,0)
Encoder 5	B5/Conv 1 B5/Conv 2 B5/Conv 3 B5/Conv 4	Image 5 B5/Conv 1 B5/Conv 2 B5/Conv 3	B5/Conv 1 B5/Conv 2 B5/Conv 3 Latent 5	$1 \times 3 \times 3 \times 5$ $1 \times 1 \times 1 \times 7$ $1 \times 1 \times 1 \times 15$ $1 \times 1 \times 1 \times 15$	(2,1) (2,1) (1,0) (1,0)
Self-expressiveness layer	Common Θ_s	Latent 1 Latent 2 Latent 3 Latent 4 Latent 5	L-recon 1 L-recon 2 L-recon 3 L-recon 4 L-recon 5	4665600 Parameters	-
Decoder 1	D1/deconv 1 D1/deconv 2 D1/deconv 3	L-recon 1 D1/deconv 1 D1/deconv 2	D1/deconv 1 D1/deconv 2 Recon 1	$1 \times 1 \times 1 \times 15$ $1 \times 1 \times 1 \times 7$ $1 \times 3 \times 3 \times 5$	(1,0) (2,1) (2,1)
Decoder 2	D2/deconv 2 D2/deconv 2 D2/deconv 3	L-recon 2 D2/deconv 1 D2/deconv 2	D2/deconv 1 D2/deconv 2 Recon 2	$1 \times 1 \times 1 \times 15$ $1 \times 1 \times 1 \times 7$ $1 \times 3 \times 3 \times 5$	(1,0) (2,1) (2,1)
Decoder 3	D3/deconv 2 D3/deconv 2 D3/deconv 3	L-recon 3 D3/deconv 1 D3/deconv 2	D3/deconv 1 D3/deconv 2 Recon 3	$1 \times 1 \times 1 \times 15$ $1 \times 1 \times 1 \times 7$ $1 \times 3 \times 3 \times 5$	(1,0) (2,1) (2,1)
Decoder 4	D4/deconv 2 D4/deconv 2 D4/deconv 3	L-recon 4 D4/deconv 1 D4/deconv 2	D4/deconv 1 D4/deconv 2 Recon 4	$1 \times 1 \times 1 \times 15$ $1 \times 1 \times 1 \times 7$ $1 \times 3 \times 3 \times 5$	(1,0) (2,1) (2,1)
Decoder 5	D5/deconv 2 D5/deconv 2 D5/deconv 3	L-recon 5 D5/deconv 1 D5/deconv 2	D5/deconv 1 D5/deconv 2 Recon 5	$1 \times 1 \times 1 \times 15$ $1 \times 1 \times 1 \times 7$ $1 \times 3 \times 3 \times 5$	(1,0) (2,1) (2,1)

TABLE XVI
MULTIMODAL DECODERS IN THE ARL EXPERIMENTS.

	Layer	Input	output	Kernel	(stride, pad)
Decoder 1	D1/deconv 1 D1/deconv 2 D1/deconv 3	L-recon D1/deconv 1 D1/deconv 2	D1/deconv 1 D1/deconv 2 Recon 1	$1 \times 1 \times 1 \times 15$ $1 \times 1 \times 1 \times 7$ $1 \times 3 \times 3 \times 5$	(1,0) (2,1) (2,1)
Decoder 2	D2/deconv 2 D2/deconv 2 D2/deconv 3	L-recon D2/deconv 1 D2/deconv 2	D2/deconv 1 D2/deconv 2 Recon 2	$1 \times 1 \times 1 \times 15$ $1 \times 1 \times 1 \times 7$ $1 \times 3 \times 3 \times 5$	(1,0) (2,1) (2,1)
Decoder 3	D3/deconv 2 D3/deconv 2 D3/deconv 3	L-recon D3/deconv 1 D3/deconv 2	D3/deconv 1 D3/deconv 2 Recon 3	$1 \times 1 \times 1 \times 15$ $1 \times 1 \times 1 \times 7$ $1 \times 3 \times 3 \times 5$	(1,0) (2,1) (2,1)
Decoder 4	D4/deconv 2 D4/deconv 2 D4/deconv 3	L-recon D4/deconv 1 D4/deconv 2	D4/deconv 1 D4/deconv 2 Recon 4	$1 \times 1 \times 1 \times 15$ $1 \times 1 \times 1 \times 7$ $1 \times 3 \times 3 \times 5$	(1,0) (2,1) (2,1)
Decoder 5	D5/deconv 2 D5/deconv 2 D5/deconv 3	L-recon D5/deconv 1 D5/deconv 2	D5/deconv 1 D5/deconv 2 Recon 5	$1 \times 1 \times 1 \times 15$ $1 \times 1 \times 1 \times 7$ $1 \times 3 \times 3 \times 5$	(1,0) (2,1) (2,1)

C. Different networks corresponding to Extended Yale-B experiments

REFERENCES

[1] R. Basri and D. W. Jacobs, "Lambertian reflectance and linear subspaces," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 25, no. 2, pp. 218–233, 2003.

[2] T. Hastie and P. Y. Simard, "Metrics and models for handwritten character recognition," *Statistical Science*, pp. 54–65, 1998.

[3] J. P. Costeira and T. Kanade, "A multibody factorization method for independently moving objects," *International Journal of Computer Vision*, vol. 29, no. 3, pp. 159–179, 1998.

[4] R. Vidal, "Subspace clustering," *IEEE Signal Processing Magazine*, vol. 28, no. 2, pp. 52–68, 2011.

[5] Y. Wu, Z. Zhang, T. S. Huang, and J. Y. Lin, "Multibody grouping via orthogonal subspace decomposition," in *null*. IEEE, 2001, p. 252.

TABLE XVII
EARLY-FUSION NETWORKS IN THE EXTENDED YALE-B EXPERIMENTS.

	Layer	Input	output	Kernel	(stride, pad)
Feature Fusion	Fusion 1	Image 1 Image 2 Image 3 Image 4 Image 5	Fusion 1	-	-
Convolutional layers	Conv 1	Fusion 1	Conv 1	$1 \times 5 \times 5 \times 10$	(2,1)
	Conv 2	Conv 1	Conv 2	$1 \times 3 \times 3 \times 20$	(2,1)
	Conv 3	Conv 2	Conv 3	$1 \times 3 \times 3 \times 30$	(1,0)
	Conv 4	Conv 3	Latent	$1 \times 3 \times 3 \times 30$	(1,0)
Self-expressiveness	Θ_s	Latent	L-recon	5914624 Parameters	-
Multimodal Decoder	Decoder layers	L-recon	Recon 1 Recon 2 Recon 3 Recon 4 Recon 5	Details in Table XXI	

TABLE XVIII
LATE-FUSION NETWORKS IN THE EXTENDED YALE-B EXPERIMENTS.

	Layer	Input	output	Kernel	(stride, pad)
Branch 1	B1/Conv 1	Image 1	B1/Conv 1	$1 \times 5 \times 5 \times 10$	(2,1)
	B1/Conv 2	B1/Conv 1	B1/Conv 2	$1 \times 3 \times 3 \times 20$	(2,1)
	B1/Conv 3	B1/Conv 2	B1/Conv 3	$1 \times 3 \times 3 \times 30$	(1,0)
	B1/Conv 4	B1/Conv 3	B1/out	$1 \times 3 \times 3 \times 30$	(1,0)
Branch 2	B2/Conv 1	Image 2	B2/Conv 1	$1 \times 5 \times 5 \times 10$	(2,1)
	B2/Conv 2	B2/Conv 1	B2/Conv 2	$1 \times 3 \times 3 \times 20$	(2,1)
	B2/Conv 3	B2/Conv 2	B2/Conv 3	$1 \times 3 \times 3 \times 30$	(1,0)
	B2/Conv 4	B2/Conv 3	B2/out	$1 \times 3 \times 3 \times 30$	(1,0)
Branch 3	B3/Conv 1	Image 3	B3/Conv 1	$1 \times 5 \times 5 \times 10$	(2,1)
	B3/Conv 2	B3/Conv 1	B3/Conv 2	$1 \times 3 \times 3 \times 20$	(2,1)
	B3/Conv 3	B3/Conv 2	B3/Conv 3	$1 \times 3 \times 3 \times 30$	(1,0)
	B3/Conv 4	B3/Conv 3	B3/out	$1 \times 3 \times 3 \times 30$	(1,0)
Branch 4	B4/Conv 1	Image 4	B4/Conv 1	$1 \times 5 \times 5 \times 10$	(2,1)
	B4/Conv 2	B4/Conv 1	B4/Conv 2	$1 \times 3 \times 3 \times 20$	(2,1)
	B4/Conv 3	B4/Conv 2	B4/Conv 3	$1 \times 3 \times 3 \times 30$	(1,0)
	B4/Conv 4	B4/Conv 3	B4/out	$1 \times 3 \times 3 \times 30$	(1,0)
Branch 5	B5/Conv 1	Image 5	B5/Conv 1	$1 \times 5 \times 5 \times 10$	(2,1)
	B5/Conv 2	B5/Conv 1	B5/Conv 2	$1 \times 3 \times 3 \times 20$	(2,1)
	B5/Conv 3	B5/Conv 2	B5/Conv 3	$1 \times 3 \times 3 \times 30$	(1,0)
	B5/Conv 4	B5/Conv 3	B5/out	$1 \times 3 \times 3 \times 30$	(1,0)
Feature Fusion	Fusion 1	B1/out B2/out B3/out B4/out B5/out	Latent	-	-
Self-expressiveness	Θ_s	Latent	L-recon	5914624 Parameters	-
Multimodal Decoder	Decoder layers	L-recon	Recon 1 Recon 2 Recon 3 Recon 4 Recon 5	Details in Table XXI	

TABLE XIX
AFFINITY FUSION NETWORKS IN THE EXTENDED YALE-B EXPERIMENTS.

	Layer	Input	output	Kernel	(stride, pad)
Encoder 1	B1/Conv 1	Image 1	B1/Conv 1	$1 \times 5 \times 5 \times 10$	(2,1)
	B1/Conv 2	B1/Conv 1	B1/Conv 2	$1 \times 3 \times 20$	(2,1)
	B1/Conv 3	B1/Conv 2	B1/Conv 3	$1 \times 3 \times 3 \times 30$	(1,0)
	B1/Conv 4	B1/Conv 3	Latent 1	$1 \times 3 \times 3 \times 30$	(1,0)
Encoder 2	B2/Conv 1	Image 2	B2/Conv 1	$1 \times 5 \times 5 \times 10$	(2,1)
	B2/Conv 2	B2/Conv 1	B2/Conv 2	$1 \times 3 \times 3 \times 20$	(2,1)
	B2/Conv 3	B2/Conv 2	B2/Conv 3	$1 \times 3 \times 3 \times 30$	(1,0)
	B2/Conv 4	B2/Conv 3	Latent 2	$1 \times 3 \times 3 \times 30$	(1,0)
Encoder 3	B3/Conv 1	Image 3	B3/Conv 1	$1 \times 5 \times 5 \times 10$	(2,1)
	B3/Conv 2	B3/Conv 1	B3/Conv 2	$1 \times 3 \times 3 \times 20$	(2,1)
	B3/Conv 3	B3/Conv 2	B3/Conv 3	$1 \times 3 \times 3 \times 30$	(1,0)
	B3/Conv 4	B3/Conv 3	Latent 3	$1 \times 3 \times 3 \times 30$	(1,0)
Encoder 4	B4/Conv 1	Image 4	B4/Conv 1	$1 \times 5 \times 5 \times 10$	(2,1)
	B4/Conv 2	B4/Conv 1	B4/Conv 2	$1 \times 3 \times 3 \times 20$	(2,1)
	B4/Conv 3	B4/Conv 2	B4/Conv 3	$1 \times 3 \times 3 \times 30$	(1,0)
	B4/Conv 4	B4/Conv 3	Latent 4	$1 \times 3 \times 3 \times 30$	(1,0)
Encoder 5	B5/Conv 1	Image 5	B5/Conv 1	$1 \times 5 \times 5 \times 10$	(2,1)
	B5/Conv 2	B5/Conv 1	B5/Conv 2	$1 \times 3 \times 3 \times 20$	(2,1)
	B5/Conv 3	B5/Conv 2	B5/Conv 3	$1 \times 3 \times 3 \times 30$	(1,0)
	B5/Conv 4	B5/Conv 3	Latent 5	$1 \times 3 \times 3 \times 30$	(1,0)
Self-expressiveness layer	Common Θ_s	Latent 1 Latent 2 Latent 3 Latent 4 Latent 5	L-recon 1 L-recon 2 L-recon 3 L-recon 4 L-recon 5	5914624 Parameters	-
Decoder 1	D1/deconv 1	L-recon 1	D1/deconv 1	$1 \times 3 \times 3 \times 30$	(1,0)
	D1/deconv 2	D1/deconv 1	D1/deconv 2	$1 \times 3 \times 3 \times 20$	(2,1)
	D1/deconv 3	D1/deconv 2	Recon 1	$1 \times 5 \times 5 \times 10$	(2,1)
Decoder 2	D2/deconv 2	L-recon 2	D2/deconv 1	$1 \times 3 \times 3 \times 30$	(1,0)
	D2/deconv 2	D2/deconv 1	D2/deconv 2	$1 \times 3 \times 3 \times 20$	(2,1)
	D2/deconv 3	D2/deconv 2	Recon 2	$1 \times 5 \times 5 \times 10$	(2,1)
Decoder 3	D3/deconv 2	L-recon 3	D3/deconv 1	$1 \times 3 \times 3 \times 30$	(1,0)
	D3/deconv 2	D3/deconv 1	D3/deconv 2	$1 \times 3 \times 3 \times 20$	(2,1)
	D3/deconv 3	D3/deconv 2	Recon 3	$1 \times 5 \times 5 \times 10$	(2,1)
Decoder 4	D4/deconv 2	L-recon 4	D4/deconv 1	$1 \times 3 \times 3 \times 30$	(1,0)
	D4/deconv 2	D4/deconv 1	D4/deconv 2	$1 \times 3 \times 3 \times 20$	(2,1)
	D4/deconv 3	D4/deconv 2	Recon 4	$1 \times 5 \times 5 \times 10$	(2,1)
Decoder 5	D5/deconv 2	L-recon 5	D5/deconv 1	$1 \times 3 \times 3 \times 30$	(1,0)
	D5/deconv 2	D5/deconv 1	D5/deconv 2	$1 \times 3 \times 3 \times 20$	(2,1)
	D5/deconv 3	D5/deconv 2	Recon 5	$1 \times 5 \times 5 \times 10$	(2,1)

TABLE XX
INTERMEDIATE SPATIAL FUSION NETWORKS IN THE EXTENDED YALE-B EXPERIMENTS.

	Layer	Input	output	Kernel	(stride, pad)
Layer 1	B1/Conv 1	Image 1	B1/Conv 1	$1 \times 5 \times 5 \times 10$	(2,1)
	B2/Conv 1	Image 2	B2/Conv 1	$1 \times 5 \times 5 \times 10$	(2,1)
	B3/Conv 1	Image 3	B3/Conv 1	$1 \times 5 \times 5 \times 10$	(2,1)
	B4/Conv 1	Image 4	B4/Conv 1	$1 \times 5 \times 5 \times 10$	(2,1)
	B5/Conv 1	Image 5	B5/Conv 1	$1 \times 5 \times 5 \times 10$	(2,1)
Feature Fusion	B23/Fusion	B2/Conv 1 B3/Conv 1	B23/Fusion	-	-
	B45/Fusion	B4/Conv 1 B5/Conv 1	B45/Fusion	-	-
Layer 2	B1/Conv 2	B1/Conv 1	B1/Conv 2	$1 \times 3 \times 3 \times 20$	(2,1)
	B23/Conv 2	B23/Fusion	B23/Conv 2	$1 \times 3 \times 3 \times 20$	(2,1)
	B45/Conv 2	B45/Fusion	B45/Conv 2	$1 \times 3 \times 3 \times 20$	(2,1)
Feature Fusion	B2345/Fusion	B23/Conv 2 B45/Conv 2	B2345/Fusion	-	-
Layer 3	B1/Conv 3	B1/Conv 2	B1/Conv 3	$1 \times 3 \times 3 \times 30$	(1,0)
	B2345/Conv 3	B2345/Fusion	B2345/Conv 3	$1 \times 3 \times 3 \times 30$	(1,0)
Feature Fusion	Ball/Fusion	B1/Conv 3 B2345/Conv 3	Ball/Fusion	-	-
Layer 4	Ball/Conv 4	Ball/Conv 3	Latent	$1 \times 3 \times 3 \times 30$	(1,0)
Self-expressiveness	Θ_s	Latent	L-recon	5914624 Parameters	-
Multimodal Decoder	Decoder layers	L-recon	Recon 1 Recon 2 Recon 3 Recon 4 Recon 5	Details in Table XXI	

[6] A. Y. Yang, J. Wright, Y. Ma, and S. S. Sastry, "Unsupervised segmentation of natural images via lossy data compression," *Computer Vision and Image Understanding*, vol. 110, no. 2, pp. 212–225, 2008.

[7] W. Hong, J. Wright, K. Huang, and Y. Ma, "Multiscale hybrid linear models for lossy image representation," *IEEE Transactions on Image Processing*, vol. 15, no. 12, pp. 3655–3671, 2006.

[8] J. Ho, M.-H. Yang, J. Lim, K.-C. Lee, and D. Kriegman, "Clustering appearances of objects under varying illumination conditions," in *null*. IEEE, 2003, p. 11.

[9] A. Goh and R. Vidal, "Segmenting motions of different types by unsupervised manifold clustering," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2007.

[10] J. Yan and M. Pollefeys, "A general framework for motion segmentation: Independent, articulated, rigid, non-rigid, degenerate and non-degenerate," in *European Conference on Computer Vision*, 2006.

[11] E. Elhamifar and R. Vidal, "Sparse subspace clustering," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2009, pp. 2790–2797.

[12] G. Liu, Z. Lin, and Y. Yu, "Robust subspace segmentation by low-rank representation," in *International Conference on Machine Learning*, 2010.

[13] P. Favaro, R. Vidal, and A. Ravichandran, "A closed form solution to robust subspace estimation and clustering," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2011.

[14] C.-G. Li, R. Vidal *et al.*, "Structured sparse subspace clustering: A unified optimization framework." in *IEEE Conference on Computer Vision and Pattern Recognition*, 2015, pp. 277–286.

[15] C. You, D. Robinson, and R. Vidal, "Scalable sparse subspace clustering by orthogonal matching pursuit," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 3918–3927.

[16] P. Ji, T. Zhang, H. Lia, M. Salzmann, and I. Reid, "Deep subspace clustering networks," in *Advances in Neural Information Processing Systems*, 2017.

[17] M. Abavisani and V. M. Patel, "Adversarial domain adaptive subspace clustering," in *IEEE International Conference on Identity, Security, and Behavior Analysis*. IEEE, 2018, pp. 1–8.

[18] M. Abavisani and V. Patel, "Domain adaptive subspace clustering." in *British Machine Vision Conference*. BMVA, 2016.

[19] C.-Y. Lu, H. Min, Z.-Q. Zhao, L. Zhu, D.-S. Huang, and S. Yan, "Robust

TABLE XXI
MULTIMODAL DECODER DETAILS IN THE EXTENDED YALE-B EXPERIMENTS.

	Layer	Input	output	Kernel	(stride, pad)
Decoder 1	D1/deconv 1	L-recon	D1/deconv 1	$1 \times 3 \times 3 \times 30$	(1,0)
	D1/deconv 2	D1/deconv 1	D1/deconv 2	$1 \times 3 \times 3 \times 20$	(2,1)
	D1/deconv 3	D1/deconv 2	Recon 1	$1 \times 5 \times 5 \times 10$	(2,1)
Decoder 2	D2/deconv 2	L-recon	D2/deconv 1	$1 \times 3 \times 3 \times 30$	(1,0)
	D2/deconv 2	D2/deconv 1	D2/deconv 2	$1 \times 3 \times 3 \times 20$	(2,1)
	D2/deconv 3	D2/deconv 2	Recon 2	$1 \times 5 \times 5 \times 10$	(2,1)
Decoder 3	D3/deconv 2	L-recon	D3/deconv 1	$1 \times 3 \times 3 \times 30$	(1,0)
	D3/deconv 2	D3/deconv 1	D3/deconv 2	$1 \times 3 \times 3 \times 20$	(2,1)
	D3/deconv 3	D3/deconv 2	Recon 3	$1 \times 5 \times 5 \times 10$	(2,1)
Decoder 4	D4/deconv 2	L-recon	D4/deconv 1	$1 \times 3 \times 3 \times 30$	(1,0)
	D4/deconv 2	D4/deconv 1	D4/deconv 2	$1 \times 3 \times 3 \times 20$	(2,1)
	D4/deconv 3	D4/deconv 2	Recon 4	$1 \times 5 \times 5 \times 10$	(2,1)
Decoder 5	D5/deconv 2	L-recon	D5/deconv 1	$1 \times 3 \times 3 \times 30$	(1,0)
	D5/deconv 2	D5/deconv 1	D5/deconv 2	$1 \times 3 \times 3 \times 20$	(2,1)
	D5/deconv 3	D5/deconv 2	Recon 5	$1 \times 5 \times 5 \times 10$	(2,1)

and efficient subspace segmentation via least squares regression," in *European Conference on Computer Vision*. Springer, 2012, pp. 347–360.

[20] P. Ji, M. Salzmann, and H. Li, "Efficient dense subspace clustering," in *IEEE Winter Conference on Applications of Computer Vision*. IEEE, 2014, pp. 461–468.

[21] E. Elhamifar and R. Vidal, "Sparse subspace clustering: Algorithm, theory, and applications," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 35, no. 11, pp. 2765–2781, 2013.

[22] G. Liu, Z. Lin, S. Yan, J. Sun, Y. Yu, and Y. Ma, "Robust recovery of subspace structures by low-rank representation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 35, no. 1, pp. 171–184, 2013.

[23] V. M. Patel, H. V. Nguyen, and R. Vidal, "Latent space sparse and low-rank subspace clustering," *IEEE Journal of Selected Topics in Signal Processing*, vol. 9, no. 4, pp. 691–701, 2015.

[24] V. M. Patel and R. Vidal, "Kernel sparse subspace clustering," in *IEEE International Conference on Image Processing*, 2014.

[25] Y. X. Wang, H. Xu, and C. Leng, "Provable subspace clustering: When lrr meets ssc," in *Advances in Neural Information Processing Systems*, C. Burges, L. Bottou, M. Welling, Z. Ghahramani, and K. Weinberger, Eds., 2013, pp. 64–72.

[26] C. Zhang, H. Fu, S. Liu, G. Liu, and X. Cao, "Low-rank tensor constrained multiview subspace clustering," in *IEEE International Conference on Computer Vision*. IEEE, 2015, pp. 1582–1590.

[27] B. Cheng, G. Liu, J. Wang, Z. Huang, and S. Yan, "Multi-task low-rank affinity pursuit for image segmentation," in *IEEE International Conference on Computer Vision*, 2011, pp. 2439–2446.

[28] K. Chaudhuri, S. M. Kakade, K. Livescu, and K. Sridharan, "Multi-view clustering via canonical correlation analysis," in *International Conference on Machine Learning*, 2009, pp. 129–136.

[29] A. Kumar, P. Rai, and H. Daume, "Co-regularized multi-view spectral clustering," in *Advances in Neural Information Processing Systems*, 2011, pp. 1413–1421.

[30] X. Zhao, N. Evans, and J. L. Dugelay, "A subspace co-training framework for multi-view clustering," *Pattern Recognition Letters*, vol. 41, pp. 73–82, 2014.

[31] K. Zhan, C. Zhang, J. Guan, and J. Wang, "Graph learning for multiview clustering," *IEEE Transactions on Cybernetics*, pp. 1–9, 2018.

[32] M. White, X. Zhang, D. Schuurmans, and Y. liang Yu, "Convex multi-view subspace learning," in *Advances in Neural Information Processing Systems*, 2012, pp. 1673–1681.

[33] H. Wang, C. Weng, and J. Yuan, "Multi-feature spectral clustering with minimax optimization," in *IEEE Conference on Computer Vision and Pattern Recognition*, June 2014, pp. 4106–4113.

[34] R. Xia, Y. Pan, L. Du, and J. Yin, "Robust multi-view spectral clustering via low-rank and sparse decomposition," in *AAAI Conference on Artificial Intelligence*, 2014, pp. 2149–2155.

[35] V. R. de Sa, "Spectral clustering with two views," in *ICML Workshop on Learning With Multiple Views*, 2005.

[36] M. Abavisani and V. M. Patel, "Multimodal sparse and low-rank subspace clustering," *Information Fusion*, vol. 39, pp. 168–177, 2018.

[37] J. Ngiam, A. Khosla, M. Kim, J. Nam, H. Lee, and A. Y. Ng, "Multimodal deep learning," in *International Conference on Machine Learning*, 2011, pp. 689–696.

[38] N. Srivastava and R. R. Salakhutdinov, "Multimodal learning with deep boltzmann machines," in *Advances in Neural Information Processing Systems*, 2012, pp. 2222–2230.

[39] D. Ramachandram and G. W. Taylor, "Deep multimodal learning: A survey on recent advances and trends," *IEEE Signal Processing Magazine*, vol. 34, no. 6, pp. 96–108, 2017.

[40] S. E. Kahou, X. Bouthillier, P. Lamblin, C. Gulcehre, V. Michalski, K. Konda, S. Jean, P. Froumenty, Y. Dauphin, N. Boulanger-Lewandowski *et al.*, "Emonets: Multimodal deep learning approaches for emotion recognition in video," *Journal on Multimodal User Interfaces*, vol. 10, no. 2, pp. 99–111, 2016.

[41] A. Jain, J. Tompson, Y. LeCun, and C. Bregler, "Modeep: A deep learning framework using motion features for human pose estimation," in *Asian conference on computer vision*. Springer, 2014, pp. 302–315.

[42] A. Valada, G. L. Oliveira, T. Brox, and W. Burgard, "Deep multispectral semantic scene understanding of forested environments using multimodal fusion," in *International Symposium on Experimental Robotics*. Springer, 2016, pp. 465–477.

[43] S. Antol, A. Agrawal, J. Lu, M. Mitchell, D. Batra, C. Lawrence Zitnick, and D. Parikh, "Vqa: Visual question answering," in *IEEE International Conference on Computer Vision*, 2015, pp. 2425–2433.

[44] J. Donahue, L. Anne Hendricks, S. Guadarrama, M. Rohrbach, S. Venugopalan, K. Saenko, and T. Darrell, "Long-term recurrent convolutional networks for visual recognition and description," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2015, pp. 2625–2634.

[45] P. Perera, M. Abavisani, and V. M. Patel, "In2i: Unsupervised multi-image-to-image translation using generative adversarial networks," *International Conference on Pattern Recognition*, 2017.

[46] X. Di and V. M. Patel, "Large margin multi-modal triplet metric learning," in *IEEE International Conference on Automatic Face and Gesture Recognition*. IEEE, 2017, pp. 370–377.

[47] D. Kiela, E. Grave, A. Joulin, and T. Mikolov, "Efficient large-scale multi-modal classification," *arXiv preprint arXiv:1802.02892*, 2018.

[48] C. Feichtenhofer, A. Pinz, and A. Zisserman, "Convolutional two-stream network fusion for video action recognition," in *IEEE Conference on Computer Vision and Pattern Recognition*. IEEE, 2016, pp. 1933–1941.

[49] A. Y. Ng, M. I. Jordan, and Y. Weiss, "On spectral clustering: Analysis and an algorithm," in *Neural Information Processing Systems*, vol. 2, 2002, pp. 849–856.

[50] S. Bickel and T. Scheffer, "Multi-view clustering." in *IEEE International Conference on Data Mining*, vol. 4, 2004, pp. 19–26.

[51] X. Cao, C. Zhang, H. Fu, S. Liu, and H. Zhang, "Diversity-induced multi-view subspace clustering," in *IEEE Conference on Computer Vision and Pattern Recognition*, June 2015, pp. 586–594.

[52] X. Cao, C. Zhang, C. Zhou, H. Fu, and H. Foroosh, "Constrained multi-view video face clustering," *IEEE Transactions on Image Processing*, vol. 24, no. 11, pp. 4381–4393, 2015.

[53] J.-H. Kim, S.-W. Lee, D. Kwak, M.-O. Heo, J. Kim, J.-W. Ha, and B.-T. Zhang, "Multimodal residual learning for visual qa," in *Advances in Neural Information Processing Systems*, 2016, pp. 361–369.

[54] N. Neverova, C. Wolf, G. Taylor, and F. Nebout, "Moddrop: adaptive multi-modal gesture recognition," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 38, no. 8, pp. 1692–1706, 2016.

[55] M. Simonovsky, B. Gutiérrez-Becker, D. Mateus, N. Navab, and N. Komodakis, "A deep metric for multimodal registration," in *International Conference on Medical Image Computing and Computer-Assisted Intervention*. Springer, 2016, pp. 10–18.

[56] S. Liu, S. Liu, W. Cai, H. Che, S. Pujol, R. Kikinis, D. Feng, M. J. Fulham *et al.*, "Multimodal neuroimaging feature learning for multiclass diagnosis of alzheimer's disease," *IEEE Transactions on Biomedical Engineering*, vol. 62, no. 4, pp. 1132–1140, 2015.

[57] Y. LeCun and C. Cortes, "Mnist handwritten digit database," *AT&T Labs [Online]. Available: http://yann. lecun. com/exdb/mnist*, 2010.

[58] J. J. Hull, "A database for handwritten text recognition research," *IEEE Transactions on pattern analysis and machine intelligence*, vol. 16, no. 5, pp. 550–554, 1994.

[59] S. Hu, N. J. Short, B. S. Riggan, C. Gordon, K. P. Gurton, M. Thielke, P. Gurram, and A. L. Chan, "A polarimetric thermal database for face recognition research," in *IEEE Conference on Computer Vision and Pattern Recognition Workshops*, 2016, pp. 119–126.

[60] K. C. Lee, J. Ho, and D. J. Kriegman, "Acquiring linear subspaces for face recognition under variable lighting," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 27, no. 5, pp. 684–698, May 2005.

[61] M. Abadi, P. Barham, J. Chen, Z. Chen, A. Davis, J. Dean, M. Devin, S. Ghemawat, G. Irving, M. Isard *et al.*, "Tensorflow: a system for

large-scale machine learning." in *Operating Systems Design and Implementation*, vol. 16, 2016, pp. 265–283.

[62] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.

[63] N. X. Vinh, J. Epps, and J. Bailey, "Information theoretic measures for clusterings comparison: Variants, properties, normalization and correction for chance," *Journal of Machine Learning Research*, vol. 11, no. Oct, pp. 2837–2854, 2010.

[64] W. M. Rand, "Objective criteria for the evaluation of clustering methods," *Journal of the American Statistical association*, vol. 66, no. 336, pp. 846–850, 1971.

[65] X. Guo, X. Liu, E. Zhu, and J. Yin, "Deep clustering with convolutional autoencoders," in *International Conference on Neural Information Processing*. Springer, 2017, pp. 373–382.

[66] S. Shekhar, V. M. Patel, N. M. Nasrabadi, and R. Chellappa, "Joint sparse representation for robust multimodal biometrics recognition," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 36, no. 1, pp. 113–126, 2014.

Mahdi Abavisani [S'11] received his M.S. degrees in Electrical and Computer Engineering (ECE) from Iran University of Science and Technology, Tehran, Iran in 2014, and Rutgers University, NJ, USA, in 2018. He is currently a Ph.D. candidate in Electrical and Computer Engineering at Rutgers University. During his Ph.D. he has spent time at Microsoft Research & AI, and Tesla's Autopilot team in designing deep neural networks for various applications. His research interests include signal and image processing, computer vision, machine learning and deep learning.

Vishal M. Patel Vishal M. Patel [SM'15] is an Assistant Professor in the Department of Electrical and Computer Engineering (ECE) at Johns Hopkins University. Prior to joining Hopkins, he was an A. Walter Tyson Assistant Professor in the Department of ECE at Rutgers University and a member of the research faculty at the University of Maryland Institute for Advanced Computer Studies (UMIACS). He completed his Ph.D. in Electrical Engineering from the University of Maryland, College Park, MD, in 2010. His current research interests include signal processing, computer vision, and pattern recognition with applications in biometrics and imaging. He has received a number of awards including the 2016 ONR Young Investigator Award, the 2016 Jimmy Lin Award for Invention, A. Walter Tyson Assistant Professorship Award, Best Paper Award at IEEE AVSS 2017, Best Paper Award at IEEE BTAS 2015, Honorable Mention Paper Award at IAPR ICB 2018, two Best Student Paper Awards at IAPR ICPR 2018, and Best Poster Awards at BTAS 2015 and 2016. He is an Associate Editor of the IEEE Signal Processing Magazine, IEEE Biometrics Compendium, and serves on the Information Forensics and Security Technical Committee of the IEEE Signal Processing Society. He is a member of Eta Kappa Nu, Pi Mu Epsilon, and Phi Beta Kappa.