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

M ANY 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.

In the case where the data consist 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 \( X = \{x_1, \ldots, x_N\} \in \mathbb{R}^{D \times N} \) be a collection of \( N \) signals \( \{x_i \in \mathbb{R}^D\}_{i=1}^N \) drawn from a union of \( N \) linear subspaces \( S_1 \cup S_2 \cup \cdots \cup S_n \) of dimensions \( \{d_1, \ldots, d_N\} \) in \( \mathbb{R}^D \). Given \( X \), the task of subspace clustering is to find sub-matrices \( X_1 \in \mathbb{R}^{D \times N_1} \), \( \ldots \), \( X_n \in \mathbb{R}^{D \times N_n} \) that lie in \( S_j \) with \( N_1 + N_2 + \cdots + N_n = N \). The sparse subspace clustering (SSC) \([21]\) and low-rank representation-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 \( C \) by solving the following optimization problem

\[
\min_C \|C\|_p + \frac{\lambda}{2}\|X - XC\|_F^2, \tag{1}
\]

where \( \|\cdot\|_p \) is the \( \ell_1 \)-norm in the case of SSC \([21]\) and the nuclear norm in the case of LRR \([22]\). Here, \( \lambda \) is a regularization parameter. In addition, to prevent the trivial solution \( C = I \), an additional constraint of \( \text{diag}(C) = 0 \) is added to the above optimization problem in the case of SSC. Once \( C \) is found, spectral clustering methods \([49]\) are applied on the affinity matrix \( W = |C| + |C|^T \) to obtain the segmentation of the data \( X \).

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

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 subspace clustering. The method optimizes an objective similar to that of \([1]\) but the matrix \( C \) is approximated using a trainable dense layer embedded within the network. Let us denote the parameters of the self-expressive layer as \( \Theta_s \). Note that these parameters are essentially the elements of \( C \) in \([1]\). The following loss function is used to train the network

\[
\min_{\Theta} \|\Theta_s\|_p + \frac{\lambda_1}{2}\|Z_{\Theta_e} - Z_{\Theta_s}\|_F^2 + \frac{\lambda_2}{2}\|X - \hat{X}_{\Theta}\|, \tag{2}
\]

s.t. \( \text{diag}(\Theta_s) = 0 \).

where \( Z_{\Theta_e} \) denotes the output of the encoder, and \( \hat{X}_{\Theta} \) is the reconstructed signal at the output of the decoder. Here, the network parameters \( \Theta \) consist of encoder parameters \( \Theta_e \), decoder parameters \( \Theta_d \) and self-expressive layer parameters \( \Theta_s \). Here, \( \lambda_1 \) and \( \lambda_2 \) 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. \([50]\) introduced an Expectation Maximization (EM) and agglomerative multiview clustering methods in \([53]\). White et al. \([52]\) 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 \([50]\), \([55]\). Kumar et al. \([29]\) proposed a co-regularization method that enforces the clusters to be aligned in different views. Zhao et al. \([50]\) use output of clustering in one view to learn discriminant
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 [31], 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 \( f : \{x_1, x_2, \ldots, x^M\} \rightarrow z \) 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 \( \mathbb{R}^{H \times W \times d_m} \), and the output has the dimension of \( \mathbb{R}^{H \times W \times d_m} \). In fact, deep network structures offer the design option for having feature maps with the same dimensions. We use \( z_{i,j,k} \) and \( x_{i,j,k}^m \) to denote the value in the spatial position \((i,j,k)\) in the output and the \( m^\text{th} \) input feature map, respectively. Various fusion functions can be used to combine the input feature maps. Below, we investigate a few.

1) Sum fusion \( z = \text{sum}(x_1, x_2, \ldots, x^M) \): computes the sum of the feature maps at the same special positions as follows

\[
z_{i,j,k} = \sum_{m=1}^{M} x_{i,j,k}^m.
\]

2) Maxpooling function \( z = \text{max}(x_1, x_2, \ldots, x^M) \): returns the maximum value of the corresponding location in the input feature maps as follows

\[
z_{i,j,k} = \text{Max}_m\{x_{i,j,k}^1,x_{i,j,k}^2,\ldots,x_{i,j,k}^M\}.
\]

3) Concatenation function \( z = \text{cat}(x_1, x_2, \ldots, x^M) \): constructs the output by concatenating the input feature maps as follows

\[
z = [x_1, x_2, \ldots, x^M],
\]

where each input has the dimension \( \mathbb{R}^{H \times W \times d_m} \) and the output has the dimension \( \mathbb{R}^{H \times W \times (d_m \times M)} \). 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 \( \{x_1^1, x_2^1, \ldots, x_N^M\} \) from \( M \) different modalities, define the corresponding data matrices as \( X^m = [x_1^m, x_2^m, \ldots, x_N^m] \), \( m \in [1,\ldots,M] \). Regardless of the network structure and the fusion function of choice, let \( \Theta_{M,e} \) denote the parameters of the multimodal encoder. Similarly, let \( \Theta_s \) be the self-expressive layer parameters and \( \Theta_{M,d} \) be the multimodal decoder parameters. Then the proposed spatial
fusion models can be trained end-to-end using the following loss function

\[
\min_{\Theta} \|\Theta_s\|_p + \frac{\lambda_1}{2} \|Z_{\Theta_{M,c}} - Z_{\Theta_{M,d}}\Theta_s\|_F^2 + \frac{\lambda_2}{2} \sum_{m=1}^{M} \|X^m - \hat{X}_m^{\Theta}\|_F^2 \\
\text{s.t.} \text{diag}(\Theta_s) = 0, \quad (6)
\]

where \(\Theta\) denotes all the training network parameters including \(\Theta_{M,c}\), \(\Theta_s\) and \(\Theta_{M,d}\). The joint representation is denoted by \(Z_{\Theta_{M,c}}\) and \(\hat{X}_m^{\Theta}\) is the reconstruction of \(X^m\). Here, \(\lambda_1\) and \(\lambda_2\) are two regularization parameters, and \(\|\cdot\|_p\) can be either \(\ell_1\) or \(\ell_2\) norm.

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

\[
W = |\Theta_s^T| + |\Theta_s^T|,
\]

where \(\Theta_s\) corresponds to the self-expressive layer weights learned by an end-to-end training strategy [16]. Thus a shared \(\Theta_s\) results in a common \(W\) across the modalities. We enforce the modalities to share a common \(\Theta_s\) 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.

---

Algorithm 1 Spatial and affinity fusion algorithms

1: procedure DMSC\((|X^m|_{m=1}^M, \lambda_1, \lambda_2, \text{\textit{mode}})^1\)
2: \textbf{if} \text{\textit{mode}} = Spatial fusion then
3: \text{Train the networks using the loss (6).}
4: \textbf{else if} \text{\textit{mode}} = Affinity fusion then
5: \text{Train the networks using the loss (7).}
6: \textbf{end if}
7: \text{Extract} \(\Theta_s\) from the trained networks.
8: \text{Normalize the columns of} \(\Theta_s\) as \(\theta^{s}_j \leftarrow \frac{\theta^{s}_j}{\|\theta^{s}_j\|_2}\).
9: \text{Form a similarity graph with} \(N\) nodes and set the weights on the edges by \(W = |\Theta_s^T| + |\Theta_s^T|\).
10: \text{Apply spectral clustering to the similarity graph.}
11: \textbf{end procedure}
12: \textbf{Output}: Segmented multimodal data.

B. End-to-End Training

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

\[
\min_{\Theta} \|\Theta_s\|_p + \frac{\lambda_1}{2} \sum_{m=1}^{M} \|Z_{\Theta_{M,c}}^m - Z_{\Theta_{M,d}}^m\Theta_s\|_F^2
\]
\[
+ \frac{\lambda_2}{2} \sum_{m=1}^{M} \|X^m - \hat{X}_m^{\Theta}\|_F^2 \quad \text{s.t. diag}(\Theta_s) = 0, \quad (7)
\]

where \(\Theta_s\) is the common self-expressive layer weights. Here, \(\lambda_1\) and \(\lambda_2\) are regularization parameters. \(Z_{\Theta_{M,c}}^m\) and \(\hat{X}_m^{\Theta}\) are respectively the latent space representation and the reconstructed decoder’s output corresponding to \(X^m\). \(\Theta^m\) denotes the network parameters corresponding to the \(m\)th modality and \(\Theta\) 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.

(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. 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].

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.

Table I

<table>
<thead>
<tr>
<th>Experiment</th>
<th>Dataset</th>
<th># of modalities</th>
<th># of samples per modality</th>
</tr>
</thead>
<tbody>
<tr>
<td>Digits</td>
<td>MNIST [57], USPS [58]</td>
<td>2</td>
<td>2000</td>
</tr>
<tr>
<td>Heterogeneous</td>
<td>ARL [59]</td>
<td>5</td>
<td>2160</td>
</tr>
<tr>
<td>Facial components</td>
<td>Extended Yale-B [60]</td>
<td>5</td>
<td>2432</td>
</tr>
</tbody>
</table>

Table II

<table>
<thead>
<tr>
<th>Structure</th>
<th>Max-pooling</th>
<th>Additive</th>
<th>Concatenation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Early fusion</td>
<td>×</td>
<td>×</td>
<td>Early-concat.</td>
</tr>
<tr>
<td>Intermediate fusion</td>
<td>Intern.-mpool.</td>
<td>Intern.-additive</td>
<td>Intern.-concat.</td>
</tr>
<tr>
<td>Late fusion</td>
<td>Late-mpool.</td>
<td>Late-additive</td>
<td>Late-concat.</td>
</tr>
</tbody>
</table>

Table III

<table>
<thead>
<tr>
<th>Dataset</th>
<th>ACC</th>
<th>NMI</th>
<th>ARI</th>
</tr>
</thead>
<tbody>
<tr>
<td>MNIST</td>
<td>92.05</td>
<td>70.1</td>
<td>67.4</td>
</tr>
<tr>
<td></td>
<td>87.07</td>
<td>80.94</td>
<td>71.64</td>
</tr>
<tr>
<td></td>
<td>84.60</td>
<td>62.33</td>
<td>57.03</td>
</tr>
<tr>
<td></td>
<td>84.60</td>
<td>62.33</td>
<td>57.03</td>
</tr>
<tr>
<td>USPS</td>
<td>72.15</td>
<td>69.9</td>
<td>37.5</td>
</tr>
<tr>
<td></td>
<td>74.73</td>
<td>80.98</td>
<td>36.61</td>
</tr>
<tr>
<td></td>
<td>65.47</td>
<td>62.41</td>
<td>28.40</td>
</tr>
</tbody>
</table>

Note that applying max-pooling and additive functions in pixel level features might result in information loss.
In our experiments, samples from all the modalities are resized to 32 × 32, and rescaled to have pixel values between 0 and 255.

\[ \text{Frobenius norm} = \| X^m - \hat{X}^m \|_F^2, \]

where \( \hat{\Theta} \) 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 \( \lambda_2 = 1 \times 10^{\frac{K}{3}} \), 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 \( \lambda_1 \) and \( \lambda_2 \) 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.

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.
### 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 intern-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 and 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...
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% 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.

![Figure 9](image)

**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.

![Figure 10](image)

**Fig. 10.** The affinity fusion method’s performance through different parameter selections for $\lambda_1$ and $\lambda_2$.

E. Regularization parameters

In this section, we analyze the sensibility of the proposed method to the regularization parameters $\lambda_1$ and $\lambda_2$ 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 $\lambda_1$s over the range of $[10^{-4}, 10^4]$. Similarly, in Figure 10(b), we fix $\lambda_1 = 1$ and this time change $\lambda_2$ in the similar range to analyze the influence of $\lambda_2$ 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
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.

| Metric | $||.||_p$ | $p < 0.3$ | $p = 0.3$ | $p = 1$ | $p = 1.5$ | $p = 2$ |
|--------|------------|-----------|-----------|--------|-----------|--------|
| PCR    | ×          | 0.22      | 0.13      | 0.17   | 0.22      |        |
| NMI    | ×          | 18.64     | 98.78     | 98.84  | 98.89     | 98.38  |
| ARI    | ×          | 02.38     | 98.20     | 98.29  | 98.38     |        |

TABLE VIII
EARLY-FUSION NETWORKS IN THE DIGITS EXPERIMENTS.

<table>
<thead>
<tr>
<th>Feature Fusion</th>
<th>Image 1</th>
<th>Image 2</th>
<th>Fusion 1</th>
<th>Kernel</th>
<th>Output</th>
</tr>
</thead>
<tbody>
<tr>
<td>Convolutional layers</td>
<td>Conv 1</td>
<td>Conv 2</td>
<td>Conv 1</td>
<td>$1 \times 1 \times 5 \times 7$</td>
<td>(2.1)</td>
</tr>
<tr>
<td>Conv 3</td>
<td>Conv 2</td>
<td>Conv 3</td>
<td>Conv 3</td>
<td>$1 \times 3 \times 3 \times 15$</td>
<td>(1.0)</td>
</tr>
<tr>
<td>Conv 4</td>
<td>Conv 3</td>
<td>Conv 3</td>
<td>Latent</td>
<td>$1 \times 1 \times 1 \times 15$</td>
<td>(1.0)</td>
</tr>
</tbody>
</table>

Self-expressiveness $\theta_s$ | L-recon | Latent | L-recon | Output |
|-----------------------------|--------|--------|--------|--------|

TABLE IX
LATE-FUSION NETWORKS IN THE DIGITS EXPERIMENTS.

<table>
<thead>
<tr>
<th>Branch 1</th>
<th>Image 1</th>
<th>Image 2</th>
<th>Fusion 1</th>
<th>Kernel</th>
<th>Output</th>
</tr>
</thead>
<tbody>
<tr>
<td>B1/Conv 1</td>
<td>B1/Conv 1</td>
<td>B1/Conv 1</td>
<td>$1 \times 1 \times 1 \times 7$</td>
<td>(2.1)</td>
<td></td>
</tr>
<tr>
<td>B1/Conv 2</td>
<td>B1/Conv 2</td>
<td>B1/Conv 2</td>
<td>$1 \times 3 \times 3 \times 15$</td>
<td>(1.0)</td>
<td></td>
</tr>
<tr>
<td>B1/Conv 3</td>
<td>B1/Conv 3</td>
<td>Latent</td>
<td>$1 \times 1 \times 1 \times 15$</td>
<td>(1.0)</td>
<td></td>
</tr>
</tbody>
</table>

Feature Fusion | B1/output | B2/output | Latent | - | - |

Self-expressiveness $\theta_s$ | L-recon | Latent | L-recon | Output |
|-----------------------------|--------|--------|--------|--------|

Multimodal Decoder | L-recon | Recon 1 | Recon 2 | Details in Table XI |

TABLE X
AFFINITY FUSION NETWORKS IN THE DIGITS EXPERIMENTS.

<table>
<thead>
<tr>
<th>Layer</th>
<th>Input</th>
<th>Output</th>
<th>Kernel</th>
<th>(stride, pad)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Encoder 1</td>
<td>B1/Conv 1</td>
<td>B1/Conv 1</td>
<td>$1 \times 1 \times 1 \times 7$</td>
<td>(2.1)</td>
</tr>
<tr>
<td>B1/Conv 2</td>
<td>B1/Conv 2</td>
<td>B1/Conv 2</td>
<td>$1 \times 3 \times 3 \times 15$</td>
<td>(1.0)</td>
</tr>
<tr>
<td>B1/Conv 3</td>
<td>B1/Conv 3</td>
<td>Latent</td>
<td>$1 \times 1 \times 1 \times 15$</td>
<td>(1.0)</td>
</tr>
</tbody>
</table>

Encoder 2 | B2/Conv 1 | B2/Conv 1 | $1 \times 1 \times 1 \times 7$ | (2.1) |
| B2/Conv 2 | B2/Conv 2 | B2/Conv 2 | $1 \times 3 \times 3 \times 15$ | (1.0) |
| B2/Conv 3 | B2/Conv 3 | Latent | $1 \times 1 \times 1 \times 15$ | (1.0) |

Self-expressiveness $\theta_s$ | Latent | L-recon | L-recon | Output |
|-----------------------------|--------|--------|--------|--------|

Decoder 1 | L-recon | D1/deconv 1 | D1/deconv 1 | $1 \times 3 \times 3 \times 15$ | (1.0) |
| D1/deconv 2 | D1/deconv 2 | D1/deconv 2 | $1 \times 1 \times 1 \times 15$ | (1.0) |

Decoder 2 | D2/deconv 1 | D2/deconv 1 | D2/deconv 1 | $1 \times 3 \times 3 \times 15$ | (1.0) |
| D2/deconv 2 | D2/deconv 2 | D2/deconv 2 | $1 \times 1 \times 1 \times 15$ | (1.0) |

TABLE XI
MULTIMODAL DECODER IN THE DIGITS EXPERIMENTS.

<table>
<thead>
<tr>
<th>Layer</th>
<th>Input</th>
<th>Output</th>
<th>Kernel</th>
<th>(stride, pad)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Decoder 1</td>
<td>D1/deconv 1</td>
<td>L-recon</td>
<td>D1/deconv 1</td>
<td>D1/deconv 2</td>
</tr>
<tr>
<td>D1/deconv 2</td>
<td>D1/deconv 2</td>
<td>D1/deconv 2</td>
<td>$1 \times 1 \times 1 \times 15$</td>
<td>(1.0)</td>
</tr>
</tbody>
</table>

Decoder 2 | D2/deconv 1 | L-recon | D2/deconv 1 | D2/deconv 2 | $1 \times 3 \times 3 \times 15$ | (1.0) |
| D2/deconv 2 | D2/deconv 2 | D2/deconv 2 | $1 \times 1 \times 1 \times 15$ | (1.0) |

TABLE XII
DIFFERENT NETWORKS CORRESPONDING TO ARL EXPERIMENTS.

A. Different networks corresponding to digits experiments

B. Different networks corresponding to ARL experiments

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.
### TABLE XII
**EARLY-FUSION NETWORKS IN THE ARL EXPERIMENTS.**

<table>
<thead>
<tr>
<th>Layer</th>
<th>Input</th>
<th>output</th>
<th>Kernel</th>
<th>stride, pad</th>
</tr>
</thead>
<tbody>
<tr>
<td>Feature Fusion</td>
<td>Fusion 1</td>
<td>Image 7</td>
<td>Fusion 1</td>
<td>-</td>
</tr>
<tr>
<td></td>
<td>Conv 1</td>
<td>Image 3</td>
<td>Image 5</td>
<td>-</td>
</tr>
</tbody>
</table>

### TABLE XIII
**LATE-FUSION NETWORKS IN THE ARL EXPERIMENTS.**

<table>
<thead>
<tr>
<th>Layer</th>
<th>Input</th>
<th>output</th>
<th>Kernel</th>
<th>stride, pad</th>
</tr>
</thead>
<tbody>
<tr>
<td>Branch 1</td>
<td>B1/Conv 1</td>
<td>Image 7</td>
<td>B1/Conv 1</td>
<td>-</td>
</tr>
<tr>
<td></td>
<td>B1/Conv 2</td>
<td>Image 2</td>
<td>B1/Conv 2</td>
<td>-</td>
</tr>
<tr>
<td></td>
<td>B1/Conv 3</td>
<td>Image 3</td>
<td>B1/Conv 3</td>
<td>-</td>
</tr>
<tr>
<td></td>
<td>B1/Conv 4</td>
<td>Image 5</td>
<td>B1/Conv 4</td>
<td>-</td>
</tr>
</tbody>
</table>

### TABLE XIV
**INTERMEDIATE SPATIAL FUSION NETWORKS IN THE ARL EXPERIMENTS.**

<table>
<thead>
<tr>
<th>Layer</th>
<th>Input</th>
<th>output</th>
<th>Kernel</th>
<th>stride, pad</th>
</tr>
</thead>
<tbody>
<tr>
<td>Feature Fusion</td>
<td>B345/Fusion</td>
<td>B4/Conv 1</td>
<td>B5/Conv 1</td>
<td>B5/Conv 1</td>
</tr>
</tbody>
</table>

### TABLE XV
**AFFINITY FUSION NETWORKS IN THE ARL EXPERIMENTS.**

<table>
<thead>
<tr>
<th>Layer</th>
<th>Input</th>
<th>output</th>
<th>Kernel</th>
<th>stride, pad</th>
</tr>
</thead>
<tbody>
<tr>
<td>Encoder 1</td>
<td>D1/Conv 1</td>
<td>Image 1</td>
<td>D1/Conv 1</td>
<td>-</td>
</tr>
<tr>
<td></td>
<td>D1/Conv 2</td>
<td>Image 2</td>
<td>D1/Conv 2</td>
<td>-</td>
</tr>
<tr>
<td></td>
<td>D1/Conv 3</td>
<td>Image 3</td>
<td>D1/Conv 3</td>
<td>-</td>
</tr>
<tr>
<td></td>
<td>D1/Conv 4</td>
<td>Image 5</td>
<td>D1/Conv 4</td>
<td>-</td>
</tr>
</tbody>
</table>

### TABLE XVI
**MULTIMODAL DECODERS IN THE ARL EXPERIMENTS.**

<table>
<thead>
<tr>
<th>Layer</th>
<th>Input</th>
<th>output</th>
<th>Kernel</th>
<th>stride, pad</th>
</tr>
</thead>
<tbody>
<tr>
<td>Decorder 1</td>
<td>D1/Decomp 1</td>
<td>L-recon 1</td>
<td>D1/Decomp 2</td>
<td>-</td>
</tr>
<tr>
<td></td>
<td>D1/Decomp 2</td>
<td>L-recon 2</td>
<td>D1/Decomp 3</td>
<td>-</td>
</tr>
<tr>
<td></td>
<td>D1/Decomp 3</td>
<td>L-recon 3</td>
<td>D1/Decomp 4</td>
<td>-</td>
</tr>
<tr>
<td></td>
<td>D1/Decomp 4</td>
<td>L-recon 4</td>
<td>D1/Decomp 5</td>
<td>-</td>
</tr>
<tr>
<td></td>
<td>D1/Decomp 5</td>
<td>L-recon 5</td>
<td>D1/Decomp 6</td>
<td>-</td>
</tr>
</tbody>
</table>

### C. Different networks corresponding to Extended Yale-B experiments

**REFERENCES**


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

<table>
<thead>
<tr>
<th>Layer</th>
<th>Input</th>
<th>output</th>
<th>Kernel</th>
<th>(stride, pad)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Feature Fusion Fusion 1</td>
<td>Image 1 Fusion 1</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>Convolutional layers Conv 1</td>
<td>Conv 1</td>
<td>1 x 3 x 3</td>
<td>1</td>
<td>(1,0)</td>
</tr>
<tr>
<td>Conv 2</td>
<td>Conv 2</td>
<td>1 x 3 x 3</td>
<td>1</td>
<td>(1,0)</td>
</tr>
<tr>
<td>Conv 3</td>
<td>Conv 3</td>
<td>1 x 3 x 3</td>
<td>1</td>
<td>(1,0)</td>
</tr>
<tr>
<td>Conv 4</td>
<td>Conv 3 Latent</td>
<td>1 x 3 x 3</td>
<td>1</td>
<td>(1,0)</td>
</tr>
<tr>
<td>Self-expressiveness $\theta_s$</td>
<td>Latest</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>Multimodal Decoder Decoder layers</td>
<td>L-recon</td>
<td>Details in Table XIX</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
TABLE XXI
MULTIMODAL DECODER DETAILS IN THE EXTENDED YALE-B EXPERIMENTS.

<table>
<thead>
<tr>
<th>Decoder</th>
<th>Layer</th>
<th>Input</th>
<th>output</th>
<th>Kernel</th>
<th>(stride, pad)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Decoder 1</td>
<td>D1/deconv 1</td>
<td>L-recon</td>
<td>D1/deconv 1</td>
<td>1 x 3 x 3 x 30</td>
<td>(1,0)</td>
</tr>
<tr>
<td>Decoder 2</td>
<td>D1/deconv 2</td>
<td>D2/deconv 1</td>
<td>D2/deconv 1</td>
<td>1 x 3 x 3 x 30</td>
<td>(2,1)</td>
</tr>
<tr>
<td>Decoder 3</td>
<td>D2/deconv 2</td>
<td>D3/deconv 1</td>
<td>D3/deconv 1</td>
<td>1 x 3 x 3 x 30</td>
<td>(1,0)</td>
</tr>
<tr>
<td>Decoder 4</td>
<td>D3/deconv 2</td>
<td>D4/deconv 1</td>
<td>D4/deconv 1</td>
<td>1 x 3 x 3 x 30</td>
<td>(2,1)</td>
</tr>
<tr>
<td>Decoder 5</td>
<td>D4/deconv 3</td>
<td>D5/deconv 2</td>
<td>D5/deconv 2</td>
<td>1 x 3 x 3 x 30</td>
<td>(1,0)</td>
</tr>
</tbody>
</table>


---

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 [V’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 ICB 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.