SPARSE AND LOW-RANK REPRESENTATION-BASED METHODS FOR MULTIMODAL CLUSTERING AND RECOGNITION

BY MAHDI ABAVISANI

A dissertation submitted to the

School of Graduate Studies

Rutgers, The State University of New Jersey

in partial fulfillment of the requirements

for the degree of

Doctor of Philosophy

Graduate Program in Electrical and Computer Engineering

Written under the direction of

Dr. Vishal M. Patel

and approved by

_____

_____

_____

_____

New Brunswick, New Jersey

January, 2021

ABSTRACT OF THE DISSERTATION

Sparse and Low-rank Representation-based Methods for Multimodal Clustering and Recognition

by Mahdi Abavisani

Dissertation Director: Dr. Vishal M. Patel

Recent advances in technology have provided massive amounts of complex high-dimensional and multimodal data for computer vision and machine learning applications. This thesis uses sparse and low-rank representation-based techniques to introduce several approaches for leveraging the complementary information from multimodal and high-dimensional data in clustering and recognition tasks. We start with a focus on subspace clustering algorithms. We extend the popular sparse and low-rank based subspace clustering methods to multimodal subspace clustering algorithms that can integrate multiple high-dimensional modalities and represent them in low-dimensional joint subspaces. We then use convolutional neural networks (CNNs) to improve our proposed multimodal subspace clustering methods and develop deep multimodal subspace clustering networks. Furthermore, we design a framework for incorporating data augmentation techniques in subspace clustering networks. In the second part of the thesis, we focus on developing multimodal classification approaches. We start with introducing deep sparse representation-based classification (DSRC) and extending it to its multimodal version. Then, we propose novel approaches for two real-world applications with high-dimensional and multimodal data. In particular, first, we introduce a method to leverage the knowledge of multiple video streams in dynamic hand gesture recognition tasks and embed the knowledge in every single unimodal network. As a result, we improve the accuracy of unimodal networks

at the test time while they remain to perform in real-time. Our second applied approach is a fusion method for combining the information in social media posts' texts and images. Both texts and images are considered high-dimensional data, and in the case of social media posts, they can sometimes be uninformative or even misleading. We presented a method that is able to filter uninformative parts of text-image pairs and leverage their complementary information to detect crisis events in social media posts. Finally, we discuss some possible future research directions.

Acknowledgements

I would like to express my sincere gratitude to my advisor, Prof. Vishal M. Patel, for his support and guidance throughout the course of my PhD. I would also like to thank the other members of my PhD committee, Profs. Zoran Gajic, Laleh Najafizadeh, Dimitris N. Metaxas for reviewing this thesis. Additionally, I would like to thank the Tesla Autopilot team, Microsoft Research and AI - Applied Science Group and Dataminr where I interned during my PhD. In particular, I'd like to thank my supervisors and mentors Dr. Andrej Karpathy and Mr. Tianjun Xiao at Tesla, Drs. Hamid Reza Vaezi Joze and Vivek Pradeep at Microsoft, and Drs. Alex Jaimes, Joel Tetreault and Shengli Hu at Dataminr as well as all of the other people who made the internships possible. I'd like to thank all of my other collaborators, teachers, and mentors who I've had the pleasure of learning from throughout my PhD studies. I thank my lab mates: Pramuditha Perera, Ahmed Alsinan, Xing Di, Poojan Oza, Vishwanath Sindagi, He Zhang, Puyang Wang, Lidan Wang, Ester Gonzalez-Sosa, Hajime Nada. Finally, I'd like to thank Mehrnoosh and all the members of my family who have encouraged and supported me over the course of this PhD.

Funding: Parts of this work were supported by the *NSF grant 1618677*, US Office of Naval Research *(ONR) Grant YIP N00014-16-1-3134* and the Northrop Grumman Mission Systems Research in Applications for Learning Machines (REALM) initiative. The proposed work in chapter 8 is done in an internship at Microsoft Research. The work in Chapter 9 is done during an internship at Dataminr.

Dedication

To my parents.

Table of Contents

List of Tables

List of Figures

Chapter 1

Introduction

Many practical applications in machine learning, computer vision, and signal processing require one to process very high-dimensional and multimodal data. Training on huge amounts of complementary data is usually considered beneficial to machine learning systems. However, high-dimensional data sources often come with irrelevant or noisy dimensions that could confuse algorithms in practice. This thesis addresses approaches that can efficiently learn and summarize the complementary information from high-dimensional and multimodal data and produce more robust systems.

High-dimensional data often lie in low-dimensional subspaces. For instance, facial images with variation in illumination [16], handwritten digits [17] and trajectories of a rigidly moving object in a video [18] are examples where low-dimensional subspaces can represent the high-dimensional data. Subspace clustering algorithms essentially use this fact to find clusters in different subspaces within a dataset [19]. 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 [19]. This problem has numerous applications in including motion segmentation [20], unsupervised image segmentation [21], image representation and compression [22] and face clustering [23].

Various methods have been developed for subspace clustering in the literature. These methods can be categorized into four main groups - algebraic methods [24, 25], iterative methods [26, 27], statistical methods [28, 29, 30], and the methods based on spectral clustering [31, 32, 33, 34, 35]. In particular, sparse and low-rank representation-based subspace clustering methods [36, 37, 38, 39] have gained a lot of interest in recent years.

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 [40, 41, 42, 43, 44, 45, 46, 47, 48, 49].

In multimodal learning problems, the idea is to use the complementary information provided by the different modalities to enhance the performance. In this thesis, we propose several methods to integrate the complementary information of multimodal data to perform efficient multimodal recognitions.

In the first part of the thesis, we use sparse and low-rank representations to develop multimodal methods for the task of subspace clustering. This includes several novel approaches based on linear, kernalized and deep neural net-based subspace clustering. In addition, in Chapter 5, we propose a method for using data augmentation in deep subspace clustering. Augmented data can in a sense be viewed as new modalities (or views) and lead to producing a more robust subspace clustering.

In the second part of the thesis, we focus on multimodal learning in the classification task. Chapter 6 extends the popular sparse representation classification (SRC) algorithm to a deep CNNs-based version. We use our deep SRC (DSRC) algorithm to develop a novel multimodal classification system in Chapter 7.

In final two chapters of this thesis, we deploy multimodal learning in two real world applications. Chapter 8 and 9 present novel multimodal classification methods for dynamic hand gesture recognition and event detection in social media.

Key contributions of this thesis can be summarized as follows:

- We present several linear and non-linear multimodal sparse and low-rank subspace clustering methods[49]:

  MSSC, a multimodal extension to the SSC [36] algorithm.

  MLRR, a multimodal extension to the LRR [37] algorithm.

  MLRSSC, a multimodal extension to the LRSSC [50] algorithm.

  KMSSC, KMLRR and KLRSSC algorithms, the kernelized versions of MSSC, MLRR and LRSSC algorithms which are able to deal with non-linear data.

  The optimization problem of the proposed linear and kernalized multimodal subspace clustering algorithms are solved using the ADMM method.

- We introduce a deep learning-based multimodal subspace clustering framework in which the self-expressiveness property is encoded in the latent space by using a fully connected layer [51].

  Novel encoder network architectures corresponding to late, early and intermediate fusion are proposed for fusing multimodal data in the task of multimodal subspace clustering.

  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.

- We introduce a framework for incorporation of data augmentation techniques in deep subspace clustering algorithms [52, 53].

  We use temporal ensembling to smooth the process of finding the subspace memberships for the randomly augmented data points.

  We propose a simple yet effective unsupervised search algorithm to automatically find the most effective augmentation policies.

- We present a transductive deep learning-based formulation for the sparse representation-based classification (SRC) method [54].

- We extend our deep sparse representation-based classification (DSRC) method to its multimodal version [55].

  In this work, we also introduce a new classification rule by ensembling the sparse codes classification rule with the predictions of a set of discriminator neural net heads.

- We present an efficient approach for leveraging the knowledge from multiple modalities in training unimodal 3D convolutional neural networks (3D-CNNs) for the task of dynamic hand gesture recognition [56, 57].

  We introduce a ”spatiotemporal semantic alignment” loss (SSA) to align the content of the features from different modalities.

In addition, we regularize this loss with our proposed "focal regularization parameter" to avoid negative knowledge transfer.

- We present a new multimodal fusion method that leverages images and texts in social media to detect crisis events [58, 59].

  We introduce a cross-attention module that can filter uninformative and misleading components from weak modalities on a sample by sample basis.

  In addition, we employ a multimodal graph-based approach to stochastically transition between embeddings of different multimodal pairs during training to regularize the learning process better and deal with limited training data constructing new matched pairs from different samples.

The following is an overview of chapters in this thesis:

Chapter 2 reviews related works in subspace clustering and sparse representation-based classification.

Chapters 3, 4 and 5 address the subspace clustering task.

In **Chapter 3**, we extend the conventional Sparse Subspace Clustering (SSC) [36], Low-rank Representation-based (LRR) [37] subspace clustering and Low-Rank Sparse Subspace Clustering (LRSSC) [50] methods for multimodal data. In our formulation, we exploit the self expressiveness property [36] of each sample in its respective modality and enforce a common coefficient matrix across the modalities. As a result, we are able to exploit the correlations as well as coupling among different modalities. Furthermore, we kernelize the proposed algorithms to handle nonlinearity in the data samples. Furthermore, the proposed optimization problems are solved using the Alternating Direction Method of Multipliers (ADMM), [60].

In **Chapter 4**, motivated by recent advances in deep multimodal learning, we propose a novel approach to the problem of multimodal subspace clustering. We present a 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. 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. We investigate three

different spatial fusion techniques based on late, early, and intermediate fusion to encode the multimodal data into a latent space. These fusion techniques are motivated by the deep multimodal learning methods in supervised learning tasks [61, 62], which 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 affinity fusion-based methods, we formulate an end-to-end training objective loss.

In **Chapter 5**, we provide a framework for using data augmentations in the training of deep subspace clustering networks. Data augmentation techniques are based on the fact that slight changes in the percept do not change the brain cognition. In classification, neural networks use this fact by applying transformations to the inputs to learn to predict the same label. However, in deep subspace clustering (DSC), the ground-truth labels are not available, and as a result, one cannot easily use data augmentation techniques. In this chapter, we propose a technique to exploit the benefits of data augmentation in DSC algorithms. We learn representations that have consistent subspaces for slightly transformed inputs. In particular, we introduce a temporal ensembling component to DSC algorithms' objective function to enable the DSC networks to maintain consistent subspaces for random transformations in the input data. Besides, we provide a simple yet effective unsupervised procedure to find efficient data augmentation policies. An augmentation policy is defined as an image processing transformation with a certain magnitude and probability of being applied to each image in each epoch. We search through the policies in a search space of the most common augmentation policies to find the best policy such that the DSC network yields the highest mean Silhouette coefficient in its clustering results on a target dataset. Our method achieves state-of-the-art performance on four standard subspace clustering datasets.

Chapters 6, 7, 8, and 9 address the classification task.

In **Chapter 6**, we present a transductive deep learning-based formulation for the sparse representation-based classification (SRC) method. The proposed network consists of a convolutional autoencoder along with a fully-connected layer. The role of the autoencoder network is to learn robust deep features for classification. On the other hand, the fully-connected layer, which is placed in between the encoder and the decoder networks, is responsible for finding

the sparse representation. The estimated sparse codes are then used for classification. Various experiments on three different datasets show that the proposed network leads to sparse representations that give better classification results than state-of-the-art SRC methods.

In **Chapter 7**, we present a deep sparse representation based fusion method for classifying multimodal signals. Our proposed model consists of multimodal encoders and decoders with a shared fully-connected layer. The multimodal encoders learn separate latent space features for each modality. The latent space features are trained to be discriminative and suitable for sparse representation. The shared fully-connected layer serves as a common sparse coefficient matrix that can simultaneously reconstruct all the latent space features from different modalities. We employ discriminator heads to make the latent features discriminative. The reconstructed latent space features are then fed to the multimodal decoders to reconstruct the multimodal signals. We introduce a new classification rule by using the sparse coefficient matrix along with the predictions of the discriminator heads. Experimental results on various multimodal datasets show the effectiveness of our method.

In **Chapter 8**, we present an efficient approach for leveraging the knowledge from multiple modalities in training unimodal 3D convolutional neural networks (3D-CNNs) for the task of dynamic hand gesture recognition. Instead of explicitly combining multimodal information, which is commonplace in many state-of-the-art methods, we propose a different framework in which we embed the knowledge of multiple modalities in individual networks so that each unimodal network can achieve improved performance. In particular, we dedicate separate networks per available modality and enforce them to collaborate and learn to develop networks with common semantics and better representations. We introduce a ”spatiotemporal semantic alignment” loss (SSA) to align the features' content from different networks. In addition, we regularize this loss with our proposed ”focal regularization parameter” to avoid negative knowledge transfer. Experimental results show that our framework improves the test time recognition accuracy of unimodal networks and provides state-of-the-art performance on various dynamic hand gesture recognition datasets.

In **Chapter 9**, we exploit multimodal learning in detecting crisis-related events in social media. Recent developments in image classification and natural language processing, coupled with the rapid growth in social media usage, have enabled fundamental advances in detecting

breaking events around the world in real-time. Emergency response is one such area that stands to gain from these advances. By processing billions of texts and images a minute, events can be automatically detected to enable emergency response workers to better assess rapidly evolving situations and deploy resources accordingly. To date, most event detection techniques in this area have focused on image-only or text-only approaches, limiting detection performance and impacting the quality of information delivered to crisis response teams. This chapter presents a new multimodal fusion method that leverages both images and texts as input. In particular, we introduce a cross-attention module that can filter uninformative and misleading components from weak modalities on a sample by sample basis. In addition, we employ a multimodal graph-based approach to stochastically transition between embeddings of different multimodal pairs during training to better regularize the learning process as well as dealing with limited training data by constructing new matched pairs from different samples. We show that our method outperforms the unimodal approaches and strong multimodal baselines by a large margin on three crisis-related tasks.

In **Chapter 10**, we provide an overview of our future work that includes a) Methods for subspace clustering of heterogeneous data, b) Using adversarial generative networks to deal with subspace clustering of heterogeneous data, c) generalizing our proposed methods to other multimodal problems such as sarcasm detection in social media posts.

Chapter 2

Background and Related Works

1. Sparse and Low-rank Subspace Clustering

1..1 Overview

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 $\mathcal{S}_1 \cup \mathcal{S}_2 \cup \cdots \cup \mathcal{S}_n$. 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 $N_1 + N_2 + \cdots + N_n = N$.

Due to their simplicity, theoretical correctness, and empirical success, subspace clustering methods that are based on *self-expressiveness property* are very popular [63]. Self-expressiveness property can be stated as

$$\mathbf{X} = \mathbf{X}\mathbf{C} \quad s.t \quad \text{diag}(\mathbf{C}) = \mathbf{0}, \tag{2.1}$$

where $\mathbf{C} \in \mathbb{R}^{N \times N}$ is the coefficient matrix. There may exist many coefficient matrices that satisfy the condition in (5.1). Among those, *subspace preserving* solutions are especially of interest to self-expressiveness based subspace clustering methods. Subspace preserving property states that if an element in $\mathbf{C}$ is non-zero, the two data points in $\mathbf{X}$ that correspond to this coefficient are in the same subspace.

Self-expressiveness based methods combine these two properties and solve a problem of the form:

$$\min_{\mathbf{C}} \mathcal{L}_{\text{S.E.}}(\mathbf{C}, \mathbf{X}) + \lambda_1 \mathcal{L}_{\text{S.P.}}(\mathbf{C}), \tag{2.2}$$

where λ_1 is a regularization constant, $\mathcal{L}_{\text{S.E.}}$ and $\mathcal{L}_{\text{S.P.}}$ impose the self-expressiveness and subspace-preserving properties, respectively. Most of the linear methods use $\mathcal{L}_{\text{S.E.}}(\mathbf{C}, \mathbf{X}) = \|\mathbf{X} - \mathbf{X}\mathbf{C}\|_F^2$. However, for $\mathcal{L}_{\text{S.P.}}(\mathbf{C})$, different methods use various regularizations, including ℓ_1-norm, ℓ_2-norm and nuclear norm [36, 63, 37].

In recent years, deep neural network-based extensions were introduced to self-expressiveness based models [64, 4, 65, 66]. For these methods, x_is do not need to be drawn from a union of linear subspaces. Instead, they use autoencoder networks to map the data points to a latent space where data points lie into a union of linear subspaces and exploit the self-expressiveness and subspace-preserving properties in the latent space. Let $\mathbf{Z} \in \mathbb{R}^{d \times N}$ be the latent space features developed by the encoder in the autoencoders. Deep subspace clustering networks solve a problem of the form:

$$\min_{\mathbf{\Theta}} \mathcal{L}_{\text{S.E.}}(\mathbf{C}, \mathbf{Z}) + \lambda_1 \mathcal{L}_{\text{S.P.}}(\mathbf{C}) + \lambda_2 \mathcal{L}_{\text{Rec.}}(\mathbf{X}, \hat{\mathbf{X}}), \tag{2.3}$$

where λ_1 and λ_2 are regularization constants, Θ is the union of trainable parameters, $\hat{\mathbf{X}}$ is the reconstruction of $\mathbf{X}$ and the output of the decoder, and $\mathcal{L}_{\text{Rec.}}(\mathbf{X}, \hat{\mathbf{X}}) = \|\mathbf{X} - \hat{\mathbf{X}}\|_F^2$ is the reconstruction loss in training the autoencoder. Once a proper $\mathbf{C}$ is found from (5.2) or (5.3), spectral clustering methods [67] are applied to the affinity matrix $\mathbf{W} = |\mathbf{C}| + |\mathbf{C}|^T$ to obtain the segmentation of the data $\mathbf{X}$.

In the following, we review some the most popular sparse and low-rank subspace clustering methods including SSC [36], LRR [37], LRSC [38] and DSC [4].

1..2 Sparse Subspace Clustering

The SSC algorithm [36], which exploits 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. Hence, SSC aims to find a sparse matrix $\mathbf{C} \in \mathbb{R}^{N \times N}$ by solving the following optimization problem

$$\min \|\mathbf{C}\|_1 \text{ s.t. } \mathbf{X} = \mathbf{X}\mathbf{C}, \ \text{diag}(\mathbf{C}) = \mathbf{0} \tag{2.4}$$

where $\|\mathbf{C}\|_1 = \sum_{i,j} |C_{i,j}|$ is the ℓ_1-norm of $\mathbf{C}$. In the case when the data is contaminated by noise and outliers, one can model the data as $\mathbf{X} = \mathbf{X}\mathbf{C} + \mathbf{N} + \mathbf{E}$, where $\mathbf{N}$ is arbitrary noise and $\mathbf{E}$ is a sparse matrix containing outliers. In this case, the following problem can be solved to estimate the sparse coefficient matrix $\mathbf{C}$

$$\min_{\mathbf{C},\mathbf{E}} \frac{\lambda}{2}\|\mathbf{X} - \mathbf{X}\mathbf{C} - \mathbf{E}\|_F^2 + \|\mathbf{C}\|_1 + \lambda_e \|\mathbf{E}\|_1$$
$$\text{s.t. } \text{diag}(\mathbf{C}) = \mathbf{0}, \tag{2.5}$$

where λ and λ_e are positive regulation parameters [68].

1..3 Low-Rank Representation-based Subspace Clustering

The LRR algorithm [37] for subspace clustering is very similar to the SSC algorithm except that a low-rank representation is found instead of a sparse representation. In particular, in the presence of noisy and occluded data, the following optimization problem is solved

$$\min_{\mathbf{C},\mathbf{E}} \frac{\lambda}{2}\|\mathbf{X} - \mathbf{X}\mathbf{C} - \mathbf{E}\|_F^2 + \|\mathbf{C}\|_* + \lambda_e\|\mathbf{E}\|_{2,1}, \tag{2.6}$$

where $\|\mathbf{C}\|_*$ is the nuclear-norm of $\mathbf{C}$ which is defined as the sum of its singular values, $\|\mathbf{E}\|_{2,1} = \sum_j \sqrt{\sum_i (E_{i,j})^2}$ is the $\ell_{2,1}$-norm of $\mathbf{E}$ and λ and λ_e are two positive regularization parameters.

1..4 Low-Rank Sparse Subspace Clustering

The representation matrix $\mathbf{C}$ can be simultaneously sparse and low-rank. Thus, LRSSC seeks to find a sparse and low-rank matrix $\mathbf{C}$ by solving the following optimization problem

$$\begin{aligned}\min_{\mathbf{C},\mathbf{E}} \frac{\lambda}{2}\|\mathbf{X} - \mathbf{X}\mathbf{C} - \mathbf{E}\|_F^2 + \|\mathbf{C}\|_1 \\ + \lambda_r\|\mathbf{C}\|_* + \lambda_e\|\mathbf{E}\|_1 \quad \text{s.t. } \operatorname{diag}(\mathbf{C}) = \mathbf{0}\end{aligned} \tag{2.7}$$

where λ, λ_r and λ_e are positive regularization parameters [50].

In SSC, LRR and LRSSC, once $\mathbf{C}$ is estimated, spectral clustering methods [69] are applied on the affinity matrix $\mathbf{W} = |\mathbf{C}| + |\mathbf{C}|^T$ to obtain the segmentation of the data $\mathbf{X}$.

1..5 Deep Subspace Clustering

The deep subspace clustering network (DSC) [4] explores the self-expressiveness property by embedding the data into a latent space using an encoder-decoder type network. Figure 4.2 gives an overview of the DSC method for unimodal subspace clustering. The method optimizes an objective similar to that of (4.1) 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 (4.1). The following loss

function is used to train the network

$$\min_{\tilde{\boldsymbol{\Theta}}} \ \|\boldsymbol{\Theta}_s\|_p + \frac{\lambda_1}{2}\|\mathbf{Z}_{\boldsymbol{\Theta}_e} - \mathbf{Z}_{\boldsymbol{\Theta}_e}\boldsymbol{\Theta}_s\|_F^2 + \frac{\lambda_2}{2}\|\mathbf{X} - \hat{\mathbf{X}}_{\tilde{\boldsymbol{\Theta}}}\|, \quad \text{s.t. } \operatorname{diag}(\boldsymbol{\Theta}_s) = \mathbf{0}, \tag{2.8}$$

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

2. Sparse Representation-based Classification

In sparse representation-based classification (SRC), given a set of labeled training samples, the goal is to classify an unseen set of test samples. Suppose that we collect all the vectorized training samples with the label i in the matrix $\mathbf{X}_{train}^i \in \mathbb{R}^{d_0 \times n_i}$, where d_0 is the dimension of each sample and n_i is the number of samples in class i, then the training matrix can be constructed as

$$\mathbf{X}_{train} = [\mathbf{X}_{train}^1, \mathbf{X}_{train}^2, \cdots, \mathbf{X}_{train}^K] \in \mathbb{R}^{d_0 \times n} \tag{2.9}$$

where $n_1 + n_2 + \cdots + n_K = n$ and we have a total of K classes.

In SRC, it is assumed that an observed sample $\mathbf{x}_{test} \in \mathbb{R}^{d_0}$ can be well approximated by a linear combination of the samples in $\mathbf{X}_{train}^i$ if $\mathbf{x}_{test}$ is from class i. Thus, it is possible to predict the class of a given unlabeled data by finding a set of samples in the training set that can better approximate $\mathbf{x}_{test}$. Mathematically, these samples can be found by solving the following optimization problem

$$\min_{\boldsymbol{\alpha}} \|\boldsymbol{\alpha}\|_0 \ \text{ s.t. } \ \mathbf{x}_{test} = \mathbf{X}_{train}\boldsymbol{\alpha}, \tag{2.10}$$

where $\|\boldsymbol{\alpha}\|_0$ counts the number of non-zero elements in $\boldsymbol{\alpha}$. The minimization problem (7.1) finds a sparse solution for the linear system. However, since the optimization problem (7.1) is an NP-hard problem, in practice, a sparsity constraint is enforced by the ℓ_1-norm of $\boldsymbol{\alpha}$ which is a convex relaxation of the above problem [70, 71]. Thus, in practice the following minimization problem is solved to obtain the sparse codes

$$\min_{\boldsymbol{\alpha}} \|\mathbf{x}_{test} - \mathbf{X}_{train}\boldsymbol{\alpha}\|_2^2 + \lambda_0\|\boldsymbol{\alpha}\|_1, \tag{2.11}$$

where λ_0 is a positive regularization parameter. Once $\boldsymbol{\alpha}$ is found, one can estimate the class label of $\mathbf{x}_{test}$ as follows

$$\text{class}(\mathbf{x}_{test}) = \arg\min_{k} \|\mathbf{x}_{test} - \mathbf{X}_{train}\delta_k(\boldsymbol{\alpha})\|_2^2, \tag{2.12}$$

where $\delta_k(\cdot)$ is the characteristic function that selects the coefficients associated with the class i.

Part I

Subspace Clustering Tasks

Chapter 3

Linear and Non-linear Multimodal Subspace Clustering

1. Introduction

In many practical computer vision and image processing applications one has to process very high-dimensional data. In practice, these high-dimensional data can be represented by a low-dimensional subspace. For instance, face images under all possible illumination conditions, handwritten digits with different variations and trajectories of a rigidly moving object in a video can all be represented by low-dimensional subspaces [16, 17, 18]. One can view the collection of data from different classes as samples from a union of low-dimensional subspaces. In subspace clustering, the objective is to find the number of subspaces, their dimensions, the segmentation of the data and a basis for each subspace [19].

Various methods have been developed for subspace clustering in the literature. These methods can be categorized into four main groups - algebraic methods [24, 25], iterative methods [26, 27], statistical methods [28, 29, 30], and the methods based on spectral clustering [31, 32, 33, 34, 35]. In particular, sparse and low-rank representation-based subspace clustering methods [36, 37, 38, 39] have gained a lot of interest in recent years.

Some of the multimodal spectral clustering and segmentation methods developed in recent years include [40, 41, 42, 43, 72, 45, 46, 47, 48]. Note that some of these algorithms use dimensionality reduction methods such as Canonical Correlation Analysis (CCA) to project the multiview data onto a low-dimensional subspace for clustering [41, 47]. Also, some of these techniques are specifically designed for two views and can not be easily generalized to multiple views [72, 48].

Various multiview sparse and low-rank representation-based subspace clustering methods have also been proposed in the literature. In particular, a multiview subspace clustering method,

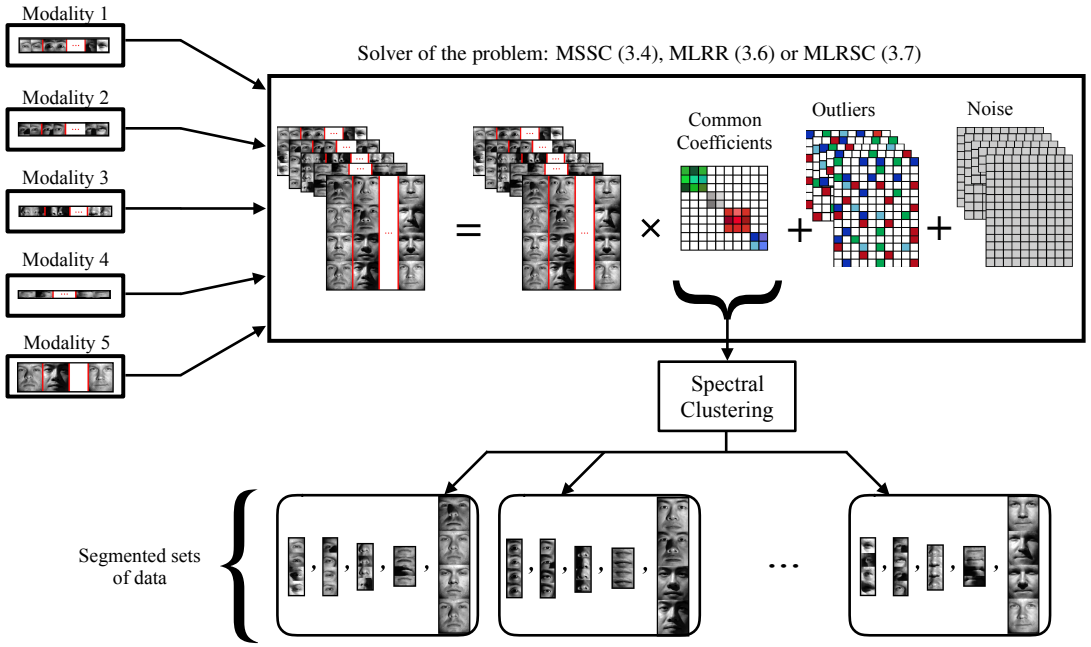


Figure 3.1: An overview of the proposed multimodal sparse and low-rank subspace clustering framework.

called Low-rank Tensor constrained Multiview Subspace Clustering (LT-MSC) was recently proposed in [73]. 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 [74], 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, [75] 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 [40]. 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.

Various supervised and unsupervised tasks and applications can get a boost in their performance by exploiting multimodal learning. This includes medical applications [76, 77, 78, 79, 80], visual recognition applications [81, 82, 83, 84, 85, 86] , computer security [87, 88], network managements [89, 90], and natural language processing tasks [91, 92, 93, 94, 95, 96].

In this chapter, we extend the Sparse Subspace Clustering (SSC) [36], Low-rank Representation-based (LRR) [37] subspace clustering and Low-Rank Sparse Subspace Clustering (LRSSC) [50] methods for multimodal data. In our formulation, we exploit the self expressiveness property [36] of each sample in its respective modality and enforce the common representation across the modalities. As a result, we are able to exploit the correlations as well as coupling among different modalities. Furthermore, we kernelize the proposed algorithms to handle nonlinearity in the data samples. The proposed optimization problems are solved using the Alternating Direction Method of Multipliers (ADMM) [60]. Figure 6.1 presents an overview of our multimodal subspace clustering framework.

This chapter is organized as follows.

Details of the proposed multimodal subspace clustering algorithms are given in Section 3.. Nonlinear extension of the proposed algorithms are presented in Section 3.. Experimental results are presented in Section 3., and finally, Section 5. concludes the chapter with a brief summary.

2. Multimodal Sparse and Low-Rank Representation-based Subspace Clustering

As discussed earlier, classical subspace clustering methods are specifically designed for unimodal data. These methods can not be easily extended to the case where we have heterogeneous data. Hence, in what follows, we present a multimodal extension of the sparse and low-rank subspace clustering algorithms. Given N paired data samples $\{(\mathbf{y}_i^1, \mathbf{y}_i^2, \cdots, \mathbf{y}_i^M)\}_{i=1}^N$ from M different modalities, define the corresponding data matrices as $\{\mathbf{Y}^m = [\mathbf{y}_1^m, \mathbf{y}_2^m, \cdots, \mathbf{y}_N^m] \in \mathbb{R}^{D_m \times N}\}_{m=1}^M$, respectively. We assume the M paired sets of sample points are drawn from a union of n linear subspaces in $\{\mathbb{R}^{D_m}\}_{m=1}^M$, respectively.

Given $\{\mathbf{Y}^m\}_{m=1}^M$, the task of multimodal subspace clustering is to simultaneously cluster the signals in distinct modalities according to their subspaces. In our formulation, we exploit the self expressiveness property of each sample in its respective modality, and enforce the common representation across the modalities.

In the case of data contaminated by noise and outliers, the data can be written as

$$\{\mathbf{Y}^m = \mathbf{Y}^m\mathbf{C}^m + \mathbf{N}^m + \mathbf{E}^m\}_{m=1}^M, \tag{3.1}$$

where $\{\mathbf{C}^m\}_{m=1}^M, \{\mathbf{N}^m\}_{m=1}^M$ and $\{\mathbf{E}^m\}_{m=1}^M$ are the corresponding sparse coefficient matrix, noise and error terms, respectively. Essentially based on this model, [73] proposed to integrate the subspace representations $\{\mathbf{C}^m\}_{m=1}^M$ using a low-rank tensor model, while [74] used a diversity induced framework to combine the representation coefficients from different modalities. Similarly, [40] proposed $\ell_{2,1}$ regularization on the concatenated subspace representations to enforce the affinities to have the consistent magnitudes. Finally, [75] proposed to minimize the distances between the normalized affinity matrices that are obtained by subspace clustering from each modality.

The key difference among the proposed method and the above mentioned methods is that in our method, the subspace representations of different modalities are enforced to be the same while in some of the previous methods, the subspace representations of different modalities are different, but somehow combined by enforcing some type of regularization (i.e. tensor, $\ell_{2,1}$, diversity links, etc.) on the representations. By extracting the common sparse and/or low-rank representation structure of data across different modalities, we are able to exploit the correlations and coupling among different modalities. As a results, we can obtain a more robust subspace sparse and/or low-rank representations. In particular, we model the data as follows

$$\{\mathbf{Y}^m = \mathbf{Y}^m\mathbf{C} + \mathbf{N}^m + \mathbf{E}^m\}_{m=1}^M, \tag{3.2}$$

where common subspace representation $\mathbf{C}$ is enforced among all modalities. Our model is motivated by [97] and [98, 99] in which common sparse representation is enforced for image super-resolution and multimodal biometrics recognition, respectively.

If the errors are sparse, then one can find $\mathbf{C}$ and $\mathbf{E} = \{\mathbf{E}^m\}_{m=1}^M$ by solving the following optimization problem

$$\begin{aligned} \min_{\mathbf{C},\mathbf{E}} \mathcal{J}(\mathbf{C},\mathbf{E}) + \frac{\lambda}{2}\sum_{m=1}^{M} \|\mathbf{Y}^m - \mathbf{Y}^m\mathbf{C} - \mathbf{E}^m\|_F^2 \\ \text{s.t. } \operatorname{diag}(\mathbf{C}) = 0. \end{aligned} \tag{3.3}$$

Depending on the choice of $\mathcal{J}$, we get different algorithms for multimodal subspace clustering. For instance, if $\mathcal{J}(\mathbf{C},\mathbf{E}) = \|\mathbf{C}\|_1 + \lambda_e\|\mathbf{E}\|_1$, we get multimodal SSC (MSSC), and the resulting

optimization problem becomes

$$\min_{\mathbf{C},\mathbf{E}} \|\mathbf{C}\|_1 + \lambda_e \|\mathbf{E}\|_1 + \frac{\lambda}{2} \sum_{m=1}^{M} \|\mathbf{Y}^m - \mathbf{Y}^m\mathbf{C} - \mathbf{E}^m\|_F^2 \tag{3.4}$$

$$\text{s.t. } \operatorname{diag}(\mathbf{C}) = 0. \tag{3.5}$$

When $\mathcal{J}(\mathbf{C},\mathbf{E}) = \|\mathbf{C}\|_* + \lambda_e\|\mathbf{E}\|_1$, we get multimodal LRR (MLRR). Note that in the case of MLRR, the term $\operatorname{diag}(\mathbf{C}) = \mathbf{0}$ in (3.3) is not required. Hence, we get the following optimization problem

$$\min_{\mathbf{C},\mathbf{E}} \|\mathbf{C}\|_* + \lambda_e \|\mathbf{E}\|_1 + \frac{\lambda}{2} \sum_{m=1}^{M} \|\mathbf{Y}^m - \mathbf{Y}^m\mathbf{C} - \mathbf{E}^m\|_F^2. \tag{3.6}$$

Finally, when $\mathcal{J}(\mathbf{C},\mathbf{E}) = \|\mathbf{C}\|_1 + \lambda_r\|\mathbf{C}\|_* + \lambda_e\|\mathbf{E}\|_1$, we get multimodal LRSSC (MLRSSC). In some cases, especially when the data is noisy, the term $\operatorname{diag}(\mathbf{C}) = \mathbf{0}$ may make the resulting representation matrix $\mathbf{C}$ not very low-rank. As a result, enforcing rank minimization along with the sparsity constraint with $\operatorname{diag}(\mathbf{C}) = \mathbf{0}$ in MLRSSC may not be that meaningful. Hence, we slightly modify the formulation in (3.3) for MLRSSC as follows

$$\min_{\mathbf{C},\mathbf{E}} \frac{\lambda}{2} \sum_{m=1}^{M} \|\mathbf{Y}^m - \mathbf{Y}^m\mathbf{A} - \mathbf{E}^m\|_F^2 + \|\mathbf{A}\|_1 + \lambda_r\|\mathbf{C}\|_* + \lambda_e\|\mathbf{E}\|_1 \quad \text{s.t. } \mathbf{A} = \mathbf{C} - \operatorname{diag}(\mathbf{C}). \tag{3.7}$$

Note that in our formulation, $\mathbf{E}$ is just a compact representation for $\{\mathbf{E}^m\}_{m=1}^M$. As will become apparent later, we solve each $\mathbf{E}^m$ separately since their dimensions may be different due to the different dimensionality of features in each modality (See Figure 6.1). Another interesting point to note here is that when $M = 1$, the proposed multimodal algorithms reduce to their unimodal counterparts.

Similar to the unimodal subspace clustering algorithms, once $\mathbf{C}$ is estimated, spectral clustering methods can be applied on the affinity matrix $\mathbf{W} = |\mathbf{C}| + |\mathbf{C}|^T$ to obtain the simultaneous segmentation of the data $\{\mathbf{Y}^m\}_{m=1}^M$. Different steps of the proposed multimodal subspace clustering algorithms are summarized in Algorithm 1.

2..1 Optimization

We present an approach based on the ADMM method [60] for solving the proposed multimodal subspace clustering problems. Due to the similarity of MSSC, MLRR and MLRSSC problems,

Algorithm 1 MSSC, MLRR, and MLRSSC Algorithms.

procedure MULTIMODAL SUBSPACE CLUSTERING($\{\mathbf{Y}^m\}_{m=1}^M$, $\lambda_e, \lambda, \lambda_r$, `'Algorithm'`)
  if `Algorithm` = MSSC **then** ▷ Obtaining $\mathbf{C}$
    Find $\mathbf{C}$ by solving (3.4).
  else if `Algorithm` = MLRR **then**
    Find $\mathbf{C}$ by solving (3.6).
  else if `Algorithm` = MLRSSC **then**
    Find $\mathbf{C}$ by solving (3.7).
  end if
  Normalize the columns of $\mathbf{C}$ as $\mathbf{c}_i \leftarrow \frac{\mathbf{c}_i}{\|\mathbf{c}_i\|_\infty}$.
  Form a similarity graph with N nodes and set the weights on the edges between the nodes by $\mathbf{W} = |\mathbf{C}| + |\mathbf{C}^T|$.
  Apply spectral clustering to the similarity graph.
end procedure
Output: Segmented multimodal data.

we only provide details on the optimization of the MSSC problem.

By introducing the auxiliary variables $\mathbf{U}$, and $\mathbf{Z}$, the MSSC problem (3.4) can be reformulated as

$$\begin{aligned} \arg\min_{\mathbf{C},\mathbf{E},\mathbf{U},\mathbf{Z}} \frac{\lambda}{2}\sum_{m=1}^{M}\|\mathbf{Y}^m - \mathbf{Y}^m\mathbf{C} - \mathbf{E}^m\|_F^2 + \|\mathbf{Z}\|_1 + \lambda_e\|\mathbf{U}\|_1 \\ \text{s.t. } \mathbf{C} = \mathbf{Z}, \mathbf{E} = \mathbf{U}, \ \ \text{diag}(\mathbf{C}) = \mathbf{0}. \end{aligned} \tag{3.8}$$

Let $f_{\alpha_C,\alpha_E}(\mathbf{C},\mathbf{E},\mathbf{Z},\mathbf{U};\mathbf{A}_E,\mathbf{A}_C)$ be the augmented Lagrangian function defined as

$$\begin{aligned} \arg\min_{\mathbf{C},\mathbf{E},\mathbf{U},\mathbf{Z}} & \frac{\lambda_n}{2}\sum_{m=1}^{M}\|\mathbf{Y}^m - \mathbf{Y}^m\mathbf{C} - \mathbf{E}^m\|_F^2 \\ & + \|\mathbf{Z}\|_1 + \frac{\alpha_C}{2}\|\mathbf{C} - (\mathbf{Z} - \text{diag}(\mathbf{Z}))\|_F^2 \\ & + \langle \mathbf{A}_C, \mathbf{C} - (\mathbf{Z} - \text{diag}(\mathbf{Z}))\rangle \\ & + \lambda_e\sum_{m=1}^{M}\|\mathbf{U}^m\|_1 + \frac{\alpha_E}{2}\sum_{m=1}^{M}\|\mathbf{E}^m - \mathbf{U}^m\|_F^2 \\ & + \sum_{m=1}^{M}\langle \mathbf{A_E}^m, \mathbf{E}^m - \mathbf{U}^m\rangle, \end{aligned} \tag{3.9}$$

where $\mathbf{A}_C$ and $\mathbf{A}_E$ are the multipliers of the constrains, α_C and α_E are positive parameters and $\langle \mathbf{A}, \mathbf{B}\rangle$ denotes trace($\mathbf{A}^T\mathbf{B}$). The resulting problem can be solved using the Augmented Lagrangian Method (ALM) [100] by keeping multipliers fixed, and updating $\mathbf{C}, \mathbf{E}, \mathbf{Z}, \mathbf{U}$, and

then updating multipliers $\mathbf{A}_C$ and $\mathbf{A}_E$ while keeping the other terms fixed. This process is repeated until convergence is reached ($\|\mathbf{C}_{k+1} - \mathbf{C}_k\|_2^F < \epsilon$).

Update step for C

Fixing $\mathbf{E}_k, \mathbf{Z}_k$, and $\mathbf{U}_k$, $\mathbf{C}_{k+1}$ can be obtained by minimizing f_{α_C,α_E} with respect to $\mathbf{C}$. Therefore, $\mathbf{C}_{k+1}$ is updated by solving the following linear system of equations

$$\left(\sum_{m=1}^{M} \lambda_n \mathbf{Y}^{mT}\mathbf{Y}^m + \alpha_C \mathbf{I}\right)\mathbf{C}_{k+1} = \left(\sum_{m=1}^{M} \lambda_n \mathbf{Y}^{mT}\left(\mathbf{Y}^m - \mathbf{E}^m\right)\right) + \alpha_C\left(\mathbf{Z}_k + \mathrm{diag}(\mathbf{Z}_k)\right) - \mathbf{A}_{C,k}, \tag{3.10}$$

where $\mathbf{I}$ is an $N \times N$ identity matrix. When N is not very large, one can simply apply matrix inversion to update $\mathbf{C}_{k+1}$ from (3.10). For large values of N, iterative methods can be used to solve (3.10) [101, 102, 103].

Update step for E

As different modalities can have features with different dimensions, $\mathbf{E}^m$s are updated separately by minimizing f_{α_C,α_E} with respect to $\mathbf{E}^m$ as follows

$$\mathbf{E}_{k+1}^m = (1+\alpha_E)^{-1}\left(\mathbf{Y}^m - \mathbf{Y}^m\mathbf{C}_{k+1} + \alpha_E\mathbf{U}_k^m - \mathbf{A}_{E,k}^m\right),$$

where $\mathbf{A}_{E,k}^m$ is the kth update of the ith modality's multiplier.

Update step for Z

The variable $\mathbf{Z}$ can be updated as follows

$$\mathbf{Z}_{k+1} = \mathbf{J} - \mathrm{diag}(\mathbf{J}),$$

where

$$\mathbf{J} \triangleq \mathcal{S}_{\frac{2}{\alpha_C}}\left(\mathbf{C}_{k+1} + \frac{2\mathbf{A}_{C,k}}{\alpha_C}\right),$$
$$\mathcal{S}_\eta(\nu) = (|\nu| - \eta)_+ + \mathrm{sgn}(\nu),$$
$$(.)_+ = \begin{cases} (|\nu| - \eta), & |\nu| - \eta \geq 0 \\ 0, & \text{Otherwise.} \end{cases}$$

Update step for U

The update step for $\mathbf{U}$ takes the following form

$$\mathbf{U}_{k+1}^{m} = \mathcal{S}_{\frac{\lambda e}{\alpha_E}} \left(\mathbf{E}_{k+1}^{m} + \alpha_E^{-1} \mathbf{A}_{E,k}^{m}, \right),$$

where $\mathcal{S}_\eta$ is the shrinkage-thresholding operator defined in the previous step.

Update steps for $\mathbf{A}_E$ and $\mathbf{A}_C$

Finally, the multipliers are updated by gradient ascent with step sizes of α_C and α_E as follows

$$\mathbf{A}_{C,k+1} = \mathbf{A}_{C,k} + \alpha_C(\mathbf{C}_{k+1} - \mathbf{Z}_{k+1})$$

$$\mathbf{A}_{E,k+1}^{m} = \mathbf{A}_{E,k}^{m} + \alpha_E(\mathbf{E}_{k+1}^{m} - \mathbf{U}_{k+1}^{m}).$$

Note that all the update steps are repeated until convergence is reached ($\|\mathbf{C}_{k+1} - \mathbf{C}_k\|_2^F < \epsilon$).

2..2 Computational Complexity

The computational complexity is an important factor in practical deployment of machine learning and computer vision algorithms. While some methods focus on optimizing low-level implementation of algorithms on hardware devices [104], it is less costly to design computationally efficient algorithms. In this section we analyze the computational complexity of the proposed multimodal subspace clustering algorithms. We denote the number of available data points in each modality as N, the dimension of multimodal features as $\{D^m\}_{m=1}^{M}$ with $D_{\text{t}} = \sum_{m=1}^{M} D^m$, and the number of subspaces as n. We also assume that the needed number of iterations to reach the convergence in solving the problems (3.4), (3.6) and (3.7) are t_1, and spectral clustering algorithm at the final step of the Algorithm 1 needs t_2 iterations.

In general, matrix multiplication of an $P \times N$ matrix with an $N \times N$ matrix has the complexity of $O\left(PN^2\right)$, and matrix addition of two $P \times N$ matrices has the complexity of $O\left(PN\right)$. In addition, both singular value decomposition (SVD) and matrix inversion of an $N \times N$ matrix has the complexity of $O\left(N^3\right)$.

The first step of the MSSC algorithm involves updating $\mathbf{C}$, which requires a matrix inversion, matrix multiplications and addiction operations. However, among the operations for updating $\mathbf{C}$, the matrix inversion with the complexity of $O\left(N^3\right)$, and the multiplications with the

Gram matrices with the computational complexities of $\{O\left(D_i N^2\right)\}_{m=1}^{M}$ can be calculated in advance, and can be used directly in the iterations.Therefore, assuming that the inverse matrix and the Gram matrices are available, updating $\mathbf{C}$ has the dominant complexity of $O\left(N^3 + D_t N^2\right)$ in each iteration. In the next step, updating each $\mathbf{E}^m$ has the dominant complexity of $O\left(D_i N^2\right)$. Updating $\mathbf{Z}$ has the complexity of $O\left(N^2\right)$ as it requires a matrix addition and thresholding each element for computing $\mathbf{J}$. Similarly, update step for $\mathbf{U}$ requires $O\left(D_t N\right)$ computations. Afterward, updating multipliers $\mathbf{A}_C$, and $\mathbf{A}_E$ have the complexities of $O\left(N^2\right)$ and $O\left(D_t N\right)$, respectively. Therefore, as the coefficient matrix is obtained after t_1 iterations, updating steps are iterated t_1 times, which results in the overall complexity of $O\left(t_1(D_t N^2 + N^3)\right)$. Finally, the spectral clustering step has the computational complexity of $O\left(t_2 n N\right)$. Therefore, the overall computational complexity of the MSSC algorithm including the inversion task at the beginning of the algorithm is $O\left(N^3 + t_1(D_t N^2 + N^3) + t_2 n N\right)$.

The computations in the MLRR and the MLRSSC algorithms are very similar to the MSSC algorithm, except that they have an additional step of the SVD where they calculate $\mathbf{Z}$. However, their dominant complexities are in the same order as with the MSSC algorithm.

3. Non-Linear Multimodal Subspace Clustering

While the linear multimodal subspace clustering models (3.4), (3.6) and (3.7) are good approximations, in practice many datasets are better modeled by non-linear manifolds. One approach to dealing with nonlinear manifolds is to use kernel methods. Kernel-based sparse representations have been exploited before in the context of sparse coding [105], dictionary learning [106], compressed sensing [107], and subspace clustering [108, 39]. It has been shown that the non-linear mapping using the kernel trick can group the data with the same distribution and make them linearly separable. In this section, we present nonlinear extensions of the proposed multimodal subspace clustering algorithms using the kernel trick.

Let $\Phi : \mathbb{R}^D \to \mathcal{H}$ be the mapping from the input space to the reproducing kernel Hilbert space $\mathcal{H}$. The kernel function $\kappa : \mathbb{R}^D \times \mathbb{R}^D \to \mathbb{R}$ is defined as the inner product $\boldsymbol{\kappa}(\mathbf{x}_i, \mathbf{x}_j) = \langle \Phi(\mathbf{x}_i), \Phi(\mathbf{x}_j) \rangle$. Then, the kernel extension of (3.3) without the sparse noise term $\mathbf{E}$ can be

formulated as

$$\min_{\mathbf{C}} \mathcal{J}(\mathbf{C}) + \frac{\lambda}{2} \sum_{m=1}^{M} \|\Phi(\mathbf{Y}^m) - \Phi(\mathbf{Y}^m)\mathbf{C}\|_F^2$$
$$\text{s.t. } \text{diag}(\mathbf{C}) = 0, \tag{3.11}$$

where $\Phi(\mathbf{Y}^m) = [\Phi(\mathbf{y}_1^m), \Phi(\mathbf{y}_2^m), \cdots, \Phi(\mathbf{y}_N^m)]$. This problem can be rewritten as

$$\min_{\mathbf{C}} \frac{\lambda}{2} \sum_{m=1}^{M} Tr(\boldsymbol{\mathcal{K}}_{\mathbf{Y}^m\mathbf{Y}^m} - 2\boldsymbol{\mathcal{K}}_{\mathbf{Y}^m\mathbf{Y}^m}\mathbf{C} + \mathbf{C}^T\boldsymbol{\mathcal{K}}_{\mathbf{Y}^m\mathbf{Y}^m}\mathbf{C}) \tag{3.12}$$
$$+ \mathcal{J}(\mathbf{C}) \text{ s.t. } \text{diag}(\mathbf{C}) = 0,$$

where $[\boldsymbol{\mathcal{K}}_{\mathbf{Y}^m\mathbf{Y}^m}]_{k,l} = [\langle \Phi(\mathbf{Y}^m), \Phi(\mathbf{Y}^m)\rangle]_{k,l} = \boldsymbol{\kappa}(\mathbf{y}_k^m, \mathbf{y}_l^m)$, and $Tr(.)$ denotes trace operation. Similar to the linear multimodal subspace clustering methods, we apply the ADMM method to efficiently solve the problem for kernel multimodal sparse and low-rank subspace clustering. We denote the nonlinear versions of MSSC, MLRR and MLRSSC as KMSSC, KMLRR and KMLRSSC, respectively.

4. Experimental Results

We evaluate the performance of our multimodal subspace clustering algorithms on five publicly available face datasets. We compare the performance of our method with several state-of-the-art subspace clustering methods such as SSC [36], LRR [34], and LRSC [35] by concatenating features from different modalities and then feeding them into these unimodal algorithms. We denote these methods as SSC-C, LRR-C and LRSC-C. In addition, we compare the performance of our method with three recently introduced state-of-the-art multimodal subspace clustering algorithms - MLAP [40], CMVFC [75], and LT-MSC [73]. Cross validation is used for parameter selection in all the experiments. Note that the MLAP algorithm requires all the modalities to have the same dimension. Therefore, the dimensions of different modalities are reduced to a common dimension (i.e. the smallest dimension among all modalities) using principal component analysis (PCA). For the experiments with the kernel multimodal subspace clustering algorithms such as KMSSC, KMLRR and KMLRSSC, we use the Gaussian kernel $\kappa(\mathbf{x}, \mathbf{y}) = \exp(-\sigma\|\mathbf{x} - \mathbf{y}\|^2)$, where σ is the parameter of the kernel function. Subspace

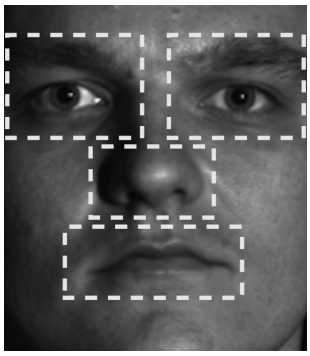


Figure 3.2: Face masks used to crop out different facial components.

Experiment	Used Features	SSC-C [36]	LRR-C [34]	LRSC-C [35]	MLAP [40]	CMVFC [75]	LT-MSC [73]	C-RP LRR
1 - Yale B Facial Components	Pixels	22.07	21.45	26.73	26.94	35.31	20.71	24.79
2 - AR Facial Components	Pixels	22.36	49.14	47.35	54.53	38.64	44.07	55.08
3 - Fusion of Features	Multiple features	25.14	20.13	25.45	23.63	34.61	18.76	22.69
4 - UMD-AA01	Alexnet "fc7"	24.49	46.35	39.35	34.69	27.73	30.62	36.73
5 - VIS NIR	Pixels	37.37	55.74	54.79	58.59	38.13	50.50	58.41
		C-RP SSC	MSSC	MLRR	MLRSSC	KMSSC	KMLRR	KMLRSSC
6 - Yale B Facial Components	Pixels	29.54	18.73	19.02	18.52	**12.67**	15.78	13.47
7 - AR Facial Components	Pixels	33.07	17.35	38.43	17.52	**10.58**	32.78	16.85
8 - Fusion of Features	Multiple features	30.82	23.36	18.61	18.83	23.25	**17.20**	18.43
9 - UMD-AA01	Alexnet "fc7"	23.12	22.45	32.56	27.11	**22.16**	26.23	27.89
10 - VIS NIR	Pixels	38.89	36.16	52.52	34.34	**30.30**	46.97	**30.30**

Table 3.1: Multimodal subspace clustering performance of different methods.

clustering error is used to measure the performance of different algorithms. It is defined as

$$\text{subspace clustering error} = \frac{\#\text{ of misclassified points}}{\text{total }\#\text{ of points}} \times 100.$$

4..1 Face Clustering using Facial Components

In the first set of experiments, we use the Extended Yale B [109], and AR face [110] datasets. We extracted four weak modalities from the face images: left and right periocular, mouth and nose regions. This was done by applying rectangular masks as shown in Figure 3.2, and cropping out the respective regions. These facial components, along with the whole face, were taken as different modalities for testing our multimodal subspace clustering methods. Simple pixel intensity values were used as features for all of them.

Subspace clustering of the Extended Yale B dataset

The Extended Yale B dataset [109] consists of 192×168 size images of 38 individuals. The dataset contains 64 frontal images of each subject under varying illumination conditions. The

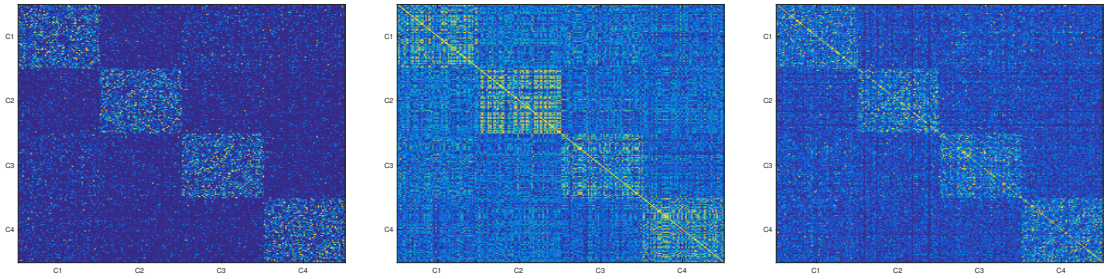
	Left Eye	Right Eye	Nose	Mouth	Face
SSC [36]	33.91	30.49	54.74	43.48	23.76
LRR [34]	**26.28**	27.39	56.46	**31.8**1	**22.52**
LRSC [35]	29.62	**25.86**	**51.93**	32.30	23.96

Table 3.2: Clustering errors on the individual facial components of the Extended Yale B dataset.

performance of SSC, LRR and LRSC on the individual facial components is summarized in Table 3.2. It can be seen from this table that among all five modalities, face gives the best performance. This is not surprising as the other modalities such as mouth, nose and eyes are considered as weak modalities, and they are not as stable as faces [111]. Overall LRR and LRSC methods seem to perform better than SSC on this dataset using individual modalities.

The first and sixth rows of Table 3.1 summarize the results obtained by different multimodal subspace cluttering methods on the Extended Yale B dataset. Once the data from different modalities are concatenated, the dimension of the resulting multimodal vector is very large. We reduce its dimension by using a random projection matrix. We denote the resulting methods as C-RP LRR and C-RP SSC. It can be seen from this table that our proposed multimodal methods perform significantly better than MLAP, CMVFC, and LT-MSC. Furthermore, it is interesting to see that the fusion results of our multimodal methods are much better than the ones obtained using single modalities. This can be clearly seen by comparing Table 3.2 with the first and sixth rows of Table 3.1. This experiment clearly shows the significance of our common sparse and low-rank representation-based methods for subspace clustering. Also, KMSSC, KMLRR and KMLRSSC further improve the performance over MSSC, MLRR and MLRSSC, respectively.

In Figure 3.3, we show the recovered common representations corresponding to the MSSC, MLRR and MLRSSC methods. Only the images from the first four subjects are used in this experiment for better visualization. As can be seen from this figure, that the recovered coefficient matrices have block diagonal structures. In particular, the coefficient matrix corresponding to the MSSC algorithm (shown in Figure 3.3 (a)) is very sparse. On the other hand, the coefficient matrix corresponding to the MLRR algorithm (shown in Figure 3.3 (b)) has many nonzero coefficients that are grouped together in a given block, which essentially corresponds to low-rankness of the common coefficient matrix. Since the MLRSSC algorithm provides a trade-off



(a) (b) (c)

Figure 3.3: Common coefficient matrices corresponding to different multimodal subspace clustering methods. Only the images from the first four subjects are used in this experiment for better visualization. $\mathbf{Ci}$ denotes coefficients of all the samples belonging to the cluster i. (a) The coefficient matrix corresponding to the MSSC algorithm. (b) The coefficient matrix corresponding to the MLRR algorithm. (c) The coefficient matrix corresponding to the MLRSSC algorithm.

between sparsity and low-rank structure of the coefficient matrix, it has more non-zero coefficients that are grouped together than the matrix corresponding to the MSSC algorithm. This can be clearly seen by comparing Figure 3.3(a) with Figure 3.3(c).

Subspace clustering of the AR face dataset

The AR face dataset [110] consists of faces from 116 individuals with varying illumination, expression and occlusion conditions, captured in two sessions. In this experiment, we choose 14 images per person from the publicly available cropped dataset[1]. These images correspond to different illumination and expression variations. The performance of unimodal methods on individual components is summarized in Table 3.3. It is interesting to see that the performance of different methods using individual components is much worse than using the entire face. This is mainly due to the fact that the AR dataset contains faces with various expressions. As a result, the weak modalities do not work well on this dataset.

The second and seventh rows of Table 3.1 summarize the results obtained by different multimodal subspace clustering methods on the AR face dataset. Although the facial components in the AR face dataset provide poor results individually, their fusion significantly enhances the performance of different subspace clustering methods. The KMSSC algorithm produces the

[1] Available at `http://www2.ece.ohio-state.edu/~aleix/ARdatabase.html`

	Left Eye	Right Eye	Nose	Mouth	Face
SSC [36]	**43.92**	**37.50**	72.78	68.07	**19.64**
LRR [34]	54.42	52.36	**61.79**	**61.21**	43.77
LRSC [35]	62.43	62.93	64.36	65.57	40.57

Table 3.3: Clustering errors on the individual facial components of the AR database.

	Pixels	LBP	Gabor	HOG	PCA
SSC [36]	23.76	41.58	33.66	27.76	24.71
LRR [34]	**22.52**	**27.31**	**20.66**	**19.05**	**18.81**
LRSC [35]	23.96	33.74	36.13	33.20	20.79

Table 3.4: Results on the Yale B dataset: clustering errors using different facial features.

best results on this dataset. Again this experiment shows the significance of our multimodal fusion method for subspace clustering. It is also interesting to note that MLRSSC algorithm provides a close performance to MSSC, but its nonlinear counterpart KMLRSSC cannot reach the performance of KMSSC. This can mainly happen because of sparse error subtraction in proposed linear methods that can significantly help satisfying low-rank constraints such as in MLRSSC.

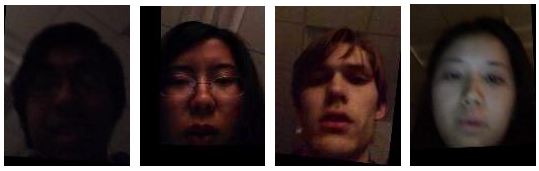
4..2 Face Clustering using Different Features

We extract different features from the face images of the Extended Yale B dataset and use them as different modalities. We extract the local binary pattern (LBP), Gabor, histogram of oriented gradients (HOG) and PCA features. Similar experiments have been conducted in [73] and [75] for face clustering.

Table 3.4 compares the performance of different subspace clustering methods on the individual features. For comparison, results corresponding to pixels are also copied from Table 3.2. This table clearly shows that extracting discriminative and robust features first and then applying subspace clustering algorithms can provide better performance over just using pixel values as features.

The results obtained by different multimodal subspace clustering methods are summarized

(a) Session one.



(b) Session two.

(c) Session three.

Figure 3.4: Sample images from different sessions in the UMD-AA01 datasets. Each session has been considered as a modality in this chapter.

in the third and eighth rows of Table 3.1. We observe that almost all methods perform much better when discriminative features are used as different modalities. Furthermore, when different features are fused using our method, their performance is significantly enhanced. Also, nonlinear kernel methods improve the performance over their linear counterparts.

Mobile Phone Facial Images Clustering

The UMD-AA01 dataset [112] is collected on mobile devices for the original purpose of active authentication, but as it contains various ambient conditions, we use it for multimodal experiments in this chapter. This dataset contains facial images of 50 users over 3 sessions corresponding to different illumination conditions. In each session more than 750 images have been taken from each face. We randomly selected seven samples per person in each session and used them in the experiments. We used the normalization method introduced in [113], then extracted deep features corresponding to the "fc7" layer from the Alexnet convolutional neural network [114]. Figure 3.4 shows some sample images from this dataset.

Table 3.5 reports the performance of various unimodal subspace clustering methods on the

	Session 1	Session 2	Session 3
SSC [36]	**37.32**	**46.36**	**40.82**
LRR [34]	41.98	47.52	48.10
LRSC [35]	44.31	48.98	44.60

Table 3.5: Clustering errors on the individual sessions of the UMD-AA01 dataset.

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

Figure 3.5: Sample images from the LDHF dataset at different standoffs (a) 1m, (b) 60m, (c) 100m and (d) 150m. Visible and near-infrared images are shown in the first and the second row, respectively.

UMD-AA01 dataset. The performance of multimodal methods is also shown in the fourth and ninth rows of Table 3.1. As can be seen from this table the use of multimodal data can improve the subspace clustering performance over their unimodal counterparts.

4..3 Visible and Infrared Face Images Clustering

In this set of experiments, we use visible and infrared faces as different modalities. Long Distance Heterogeneous Face Database (LDHF) database [115] consists of visible and near-infrared face images of 100 individuals (70 males and 30 females). The face images were captured in both daytime and nighttime at different standoffs (e.g., 1m, 60m 100m, and150m) resulting in four VIS-NIR pairs per subject. Sample image pairs from this dataset are shown in Figure 3.5. In this experiment, the face area is cropped and resized to a fixed size of 100×100 pixels. We simply use the pixel intensities as features.

Results corresponding to different unimodal subspace clustering methods are reported in Table 3.6. It can be seen from the table that generally visible images provide better performance

	Visible	Near-infrared
SSC [36]	**42.17**	**49.49**
LRR [34]	57.45	61.44
LRSC [35]	58.83	60.85

Table 3.6: Results on VIS-NIR: clustering errors using visible and near infrared images.

in terms of clustering error. In addition, this table shows that LRR has a poor performance on this dataset. This can be explained by the fact that in this dataset, we are dealing with too many number of subjects with a few samples from each subject. It has been observed that increasing the number of subjects makes subspace clustering difficult [36].

The fifth and tenth rows of the Table 3.1 provide clustering errors of multimodal subspace clustering methods on the VIS-NIR dataset. We can observe that the proposed MSSC, MLRSSC, and their kernel extensions provide the best results. An interesting observation from the Table 3.1 is that the LT-MSC method, which is a linear low-rank representation-based method, has a slightly better performance on the VIS-NIR dataset compared to the MLRR method. Similar trend is also observed on the other datasets as well. However, it should be noted that the LT-MSC needs m more parameters to select for balancing the representations from the m modalities. While this is not the case in our MLRR method. It is interesting to note that the MLRR and KMLRR algorithms do provide significant improvements over the unimodal LRR method and the other low-rank representation-based methods.

The fact that low-rank representation-based methods in this experiment are showing weaker performances compared to the sparsity-based methods can be explained by the fact that there are a large number of subjects and low number of samples per subject in VIS-NIR dataset. Figure 3.6 shows the first 12 largest singular values corresponding to one subject's data in the Extended Yale B, AR, session one in UMD-AA01 and VIS datasets. It is clear from this figure that samples in all four datasets do lie in a lower dimensional subspaces since the singular values drop quickly. In particular, each subject in the Extended Yale B dataset, AR dataset, UMD-AA01 dataset and VIS dataset, correspond to a subspace of dimension 9, 4, 4 and 3, respectively. However, considering the number of samples in each cluster one can see VIS dataset cannot show a low-rank structure as much as other datasets can show.

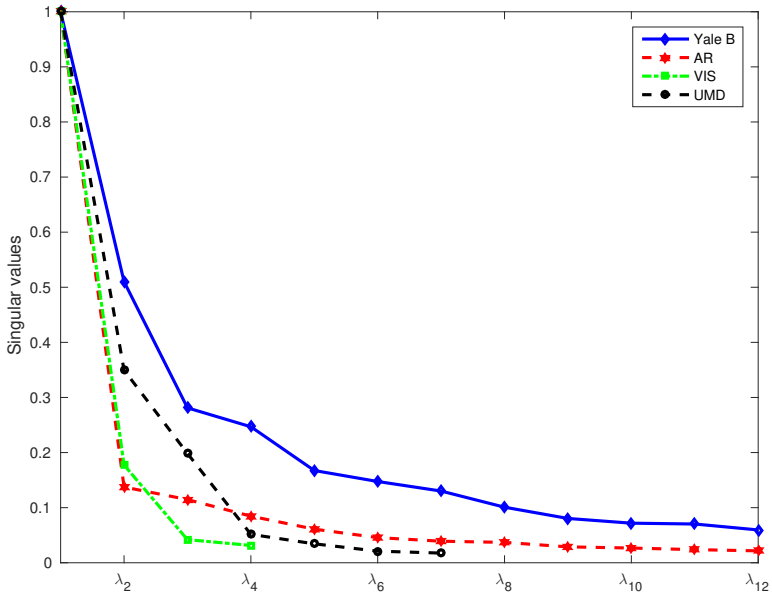


Figure 3.6: First largest singular values of samples corresponding to the first person in Yale B, AR, session one in UMD-AA01 and VIS datasets.

4..4 Impact of illumination variation

In this section, we compare the effect of illumination variations on the performance of different multimodal subspace clustering methods. We split the Yale B dataset according to different illumination variations. We choose one of the images per subject as a reference image, and the other images will be divided into four subsets according to the light angle difference from the reference. Figure 3.7 shows the variation within different subsets. We apply the same rectangular masks shown in Figure 3.2 for extracting the facial components.

Table 3.7 compares the performances of various methods on the different subsets. As expected, as illumination variations become intense, the performance of different methods drop significantly. It is interesting to see that the nonlinear methods show less dependency on the amount of variations in the sample sets. This is because kernel methods can find non-linear relations between the samples, while linear methods cannot easily deal with these variations.

4..5 Runtime comparisons

In order to compare the computational complexity of different multimodal subspace clustering methods, we measure the running time of different algorithms. Since all the compared and

(a) Subset 1.

(b) Subset 2.

(c) Subset 3.

(c) Subset 4.

Figure 3.7: Illumination variation within the selected subsets in the Yale B dataset.

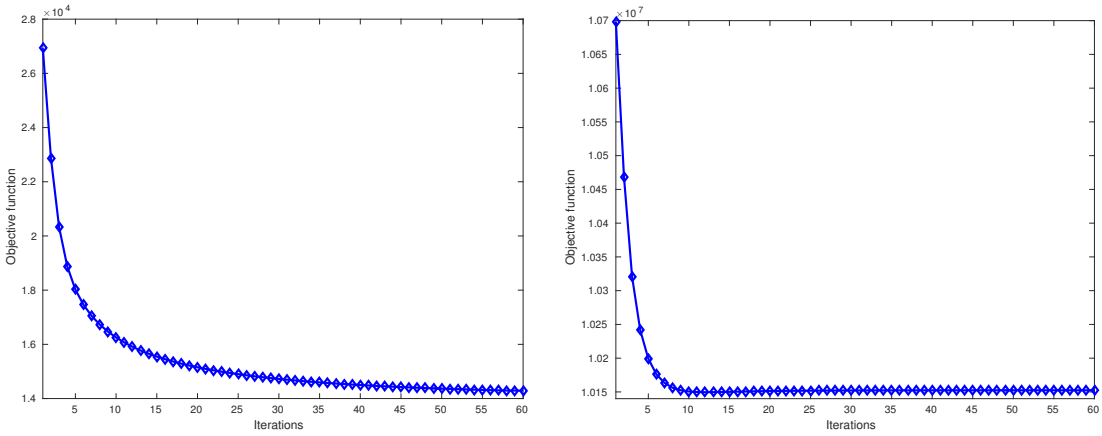
	SSC-C [36]	LRR-C [34]	LRSC-C [35]	MLAP [40]	CMVFC [75]	LT-MSC [73]	C-RP LRR
1 - Subset 1	19.55	36.27	21.80	21.99	12.40	25.43	23.30
2 - Subset 2	37.97	46.99	25.94	24.24	18.47	34.98	28.57
3 - Subset 3	39.47	70.11	61.27	56.39	33.64	52.31	60.52
4 - Subset 4	43.23	74.06	66.72	60.33	34.39	56.52	65.03
	C-RP SSC	MSSC	MLRR	MLRSSC	KMSSC	KMLRR	KMLRSSC
5 - Subset 1	18.23	9.58	21.42	8.76	8.22	18.98	**6.39**
6 - Subset 2	34.86	16.35	23.49	16.13	**13.27**	22.34	14.47
7 - Subset 3	36.53	28.57	48.49	27.43	24.97	25.43	**23.49**
8 - Subset 4	45.48	33.83	54.50	32.71	31.22	36.27	**31.07**

Table 3.7: Multimodal subspace clustering performance of different methods vs illumination variation in the data points of the Yale B face dataset.

proposed methods are iterative algorithms, many factors such as step size of gradient descent, maximum number of iterations and choice of regulation parameters can affect their running time. Thus, we report the running time of the experiments on a specific dataset. In particular, we measure the runtime of the methods on the UMD-AA01 dataset with the same settings that resulted in the reported clustering errors in the fourth and ninth rows of Table 3.1. For the methods with publicly available software packages, we use their published codes. Regarding the nonlinear methods and random projection methods, calculations of finding Gram matrices and extracting the projected features are also included in the reported runtimes. Besides, each experiment is conducted 10 times, and the average runtime is reported. All the simulations were done in Matlab on an Intel® Xeon(R) 16-core machine with 3.0 GHz CPU and 32 GB RAM, running Linux Ubuntu 14.04. Table 3.8 compares the runtime time of different methods.

Method:	SSC-C [36]	LRR-C [34]	LRSC-C [35]	MLAP [40]	CMVFC [75]	LT-MSC [73]	C-RP LRR
Time (Seconds)	45.05	3.85	**0.29**	20.49	167.42	1.18	1.72
Method:	C-RP SSC	MSSC	MLRR	MLRSSC	KMSSC	KMLRR	KMLRSSC
Time (Seconds)	36.01	16.26	1.63	2.30	3.73	23.20	2.66

Table 3.8: Runtime of different multimodal subspace clustering algorithms on the UMD-AA01 dataset.



(a) (b)

Figure 3.8: Objective function of proposed algorithms versus iterations. (a) Convergence plot of the MSSC algorithm. (b) Convergence plot of the KMSSC algorithm.

As can be seen from this table, the proposed methods are computationally efficient compared to some of the other subspace clustering methods.

4..6 Convergence

To empirically show the convergence of our method, in Figure 3.8 (a) and (b), we show the objective function vs iteration plots of the ADMM method for solving the MSSC and KMSSC problems, respectively with the experiments on the AR dataset. As can be seen from this figure, the proposed algorithms do converge in a few iterations. Experiments have shown that the MLRR, KMLRR, MLRSSC and KMLRSSC algorithms also converge in a few iterations.

5. Conclusion

We introduced multimodal extensions of the classical SSC, LRR and LRSSC methods for subspace clustering. The proposed optimization algorithms are efficiently solved using the ADMM

method. Furthermore, using the kernel trick, we made the proposed multimodal subspace clustering methods nonlinear. Extensive experiments on face clustering using publicly available datasets showed that the proposed methods can perform better than many state-of-the-art multimodal subspace clustering methods.

Chapter 4

Deep Multimodal Subspace Clustering

1. 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 [16], handwritten digits [17] and trajectories of a rigidly moving object in a video [18] 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 [19]. 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 [19]. This problem has numerous applications in including motion segmentation [20], unsupervised image segmentation [21], image representation and compression [22] and face clustering [23].

Various subspace clustering methods have been proposed in the literature [31, 32, 33, 34, 35, 116, 117, 4, 118, 119]. In particular, methods based on sparse and low-rank representation have gained a lot of attraction in recent years [36, 37, 116, 117, 39, 108, 38, 73]. 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 [4] 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 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 [40, 41, 42, 43, 44, 45, 46, 47, 48, 49]. 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 [49].

Motivated by the recent advances in deep subspace clustering [4] as well as multimodal data processing using CNNs [120, 121, 122, 84, 85, 86, 91, 92, 95], in this chapter, 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 4.1 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 [61, 62], 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

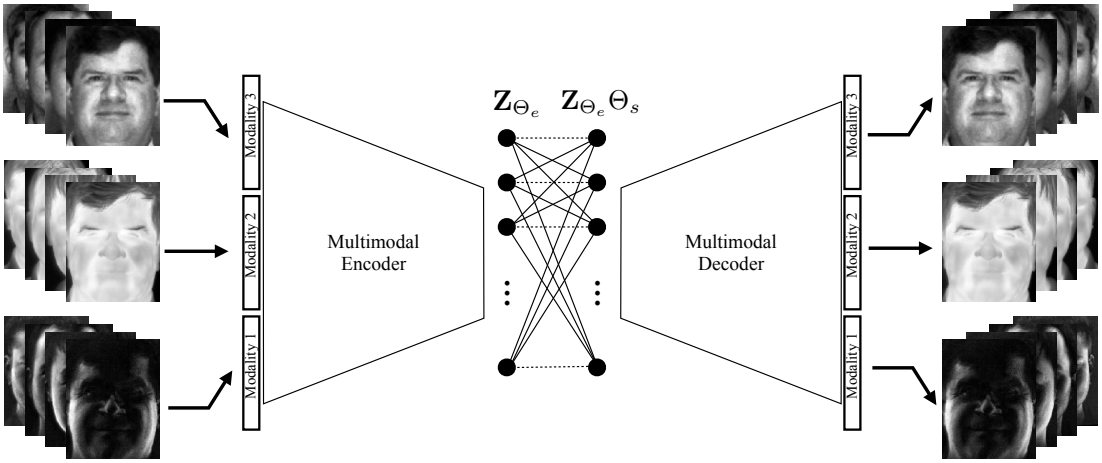


Figure 4.1: An overview of the proposed deep multimodal subspace clustering framework. Note that the network consists of three blocks: a multimodal encoder, a self-expressive layer, and a multimodal decoder. The weights in the self-expressive layer, $\mathbf{\Theta}_s$, are used to construct the affinity matrix. We present several models for the multimodal encoder.

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 chapter is organized as follows. Related works on subspace clustering and multimodal learning are presented in Section 2.. The proposed spatial fusion-based and affinity fusion-based multimodal subspace clustering methods are presented in Section 3. and 4., respectively. Experimental results are presented in Section 5., and finally, Section 6. concludes the chapter with a brief summary.

2. Related Work

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

2..1 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) [36] and low-rank representations-based subspace clustering (LRR) [37] 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

$$\min_{\mathbf{C}} \|\mathbf{C}\|_p + \frac{\lambda}{2}\|\mathbf{X} - \mathbf{X}\mathbf{C}\|_F^2, \tag{4.1}$$

where $\|.\|_p$ is the ℓ_1-norm in the case of SSC [36] and the nuclear norm in the case of LRR [37]. 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 [67] 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 [39, 108].

2..2 Deep Subspace Clustering

The deep subspace clustering network (DSC) [4] explores the self-expressiveness property by embedding the data into a latent space using an encoder-decoder type network. Figure 4.2 gives an overview of the DSC method for unimodal subsapce clustering. The method optimizes an objective similar to that of (4.1) 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 (4.1). The following loss function is used to train the network

$$\min_{\tilde{\mathbf{\Theta}}} \|\mathbf{\Theta}_s\|_p + \frac{\lambda_1}{2}\|\mathbf{Z}_{\mathbf{\Theta}_e} - \mathbf{Z}_{\mathbf{\Theta}_e}\mathbf{\Theta}_s\|_F^2 + \frac{\lambda_2}{2}\|\mathbf{X} - \hat{\mathbf{X}}_{\tilde{\mathbf{\Theta}}}\|,$$
$$\text{s.t. } \text{diag}(\mathbf{\Theta}_s) = \mathbf{0}, \tag{4.2}$$

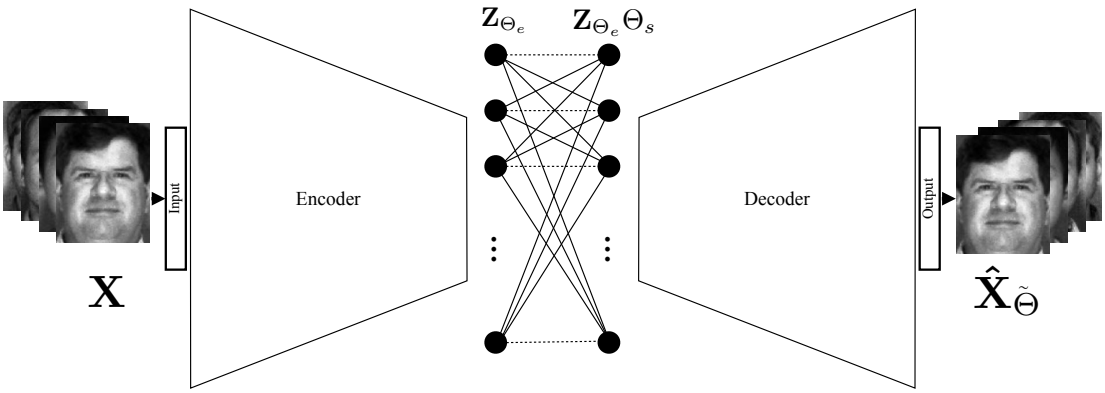


Figure 4.2: An overview of the DSC framework proposed in [4] for unimodal subspace clustering.

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.

2..3 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 [46]. White *et al.* [45] 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 [41, 47]. Some other multimodal methods are specifically designed for two views and can not be easily generalized to multiple views [72, 48]. Kumar *et al.* [42] proposed a co-regularization method that enforces the clusterings to be aligned in different views. Zhao *et al.* [43] 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 [73]. 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 [74], a diversity-induced multiview subspace clustering was proposed in which

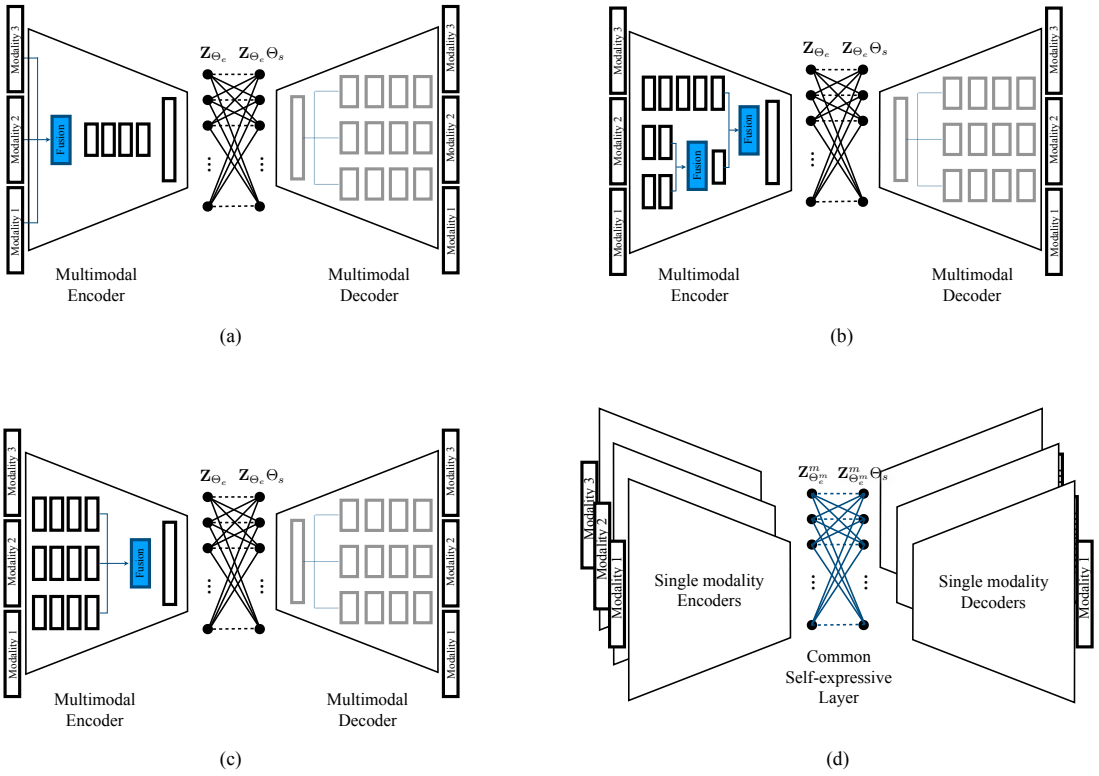


Figure 4.3: Different network architectures corresponding to (a) early fusion, (b) intermediate fusion, and (c) late fusion. Note that in all the spatial fusion-based networks (a)-(c), the overall structure for the self-expressive layer and the multimodal decoder remain the same. (d) Network architecture 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.

the Hilbert Schmidt independence criterion was utilized to explore the complementarity of multiview representations. Recently, [75] 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 [40]. 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.

2..4 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

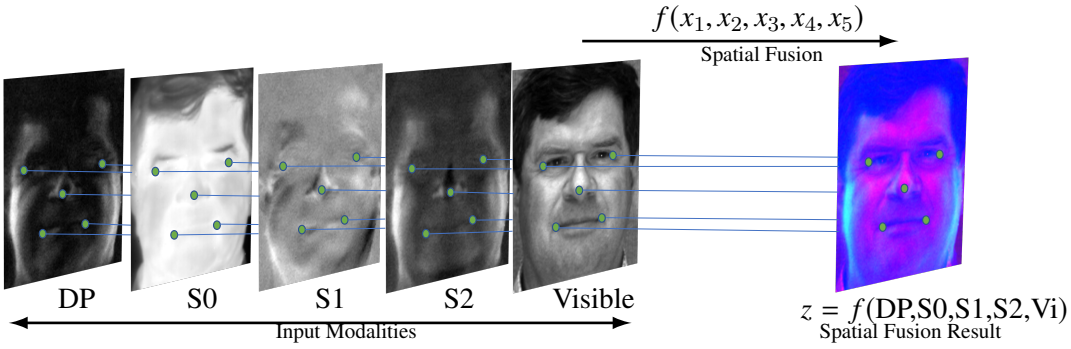


Figure 4.4: In spatial fusion methods each location of the fusion is related to the input values at the same location. In this especial case, the facial components (i.e. eyes, nose and mouth) are aligned across all the modalities (i.e. DP, S0, S1, S2, Visible).

learning was first introduced in [120], [121], and has gained a lot of attention in recent years [96, 81, 84].

Keila *et al.* [61] 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.* [62] 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 [78, 79, 123] visual recognition [85, 84] and visual question answering [96, 91]. We refer readers to [122] 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.

Various supervised and unsupervised tasks and applications can get a boost in their performance by exploiting multimodal learning. This includes medical applications [76, 77, 78, 79, 124, 80], visual recognition applications [81, 82, 83, 84, 85, 86] , computer security [125, 88, 126], network managements [127], and natural language processing tasks [91, 92, 93, 94, 95,

96].

3. 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 4.4 shows a visual example of spatial fusion where five different modalities (DP, S0, S1, S2, Visible) are combined to produce a fused result Y. The spatial fusion methods are especially popular in supervised multimodal learning applications [61, 62]. 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.

3..1 Fusion Structures

We build our deep multimodal subspace clustering networks based on the architecture proposed in [4] 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 4.3 (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 [122].

3..2 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 $f : \{x^1, x^2. \cdots, x^M\} \rightarrow z$ 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 $\mathbb{R}^{H\times W\times d^{in}}$, 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.

Sum fusion $z = \mathbf{sum}(x^1, x^2. \cdots, x^M)$

computes the sum of the feature maps at the same special positions as follows

$$z_{i,j,k} = \sum_{m=1}^{M} x^m_{i,j,k}. \tag{4.3}$$

Maxpooling function $z = \mathbf{max}(x^1, x^2. \cdots, x^M)$

returns the maximum value of the corresponding location in the input feature maps as follows

$$z_{i,j,k} = \mathrm{Max}\{x^1_{i,j,k}, x^2_{i,j,k}. \cdots, x^M_{i,j,k}\}. \tag{4.4}$$

Concatenation function $z = \textbf{cat}(x^1, x^2. \cdots, x^M)$

constructs the output by concatenating the input feature maps as follows

$$z = [x^1, x^2. \cdots, x^M], \tag{4.5}$$

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 4.3 (a)-(c).

3..3 End-to-End Training Objective

Given N paired data samples $\{\mathbf{x}_i^1, \mathbf{x}_i^2, \cdots, \mathbf{x}_i^M\}_{i=1}^N$ from M different modalities, define the corresponding data matrices as $\mathbf{X}^m = [\mathbf{x}_1^m, \mathbf{x}_s^m, \cdots, \mathbf{x}_N^m], \quad m \in \{1, \cdots, M\}$. 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

$$\min_{\mathbf{\Theta}} \|\mathbf{\Theta}_s\|_p + \frac{\lambda_1}{2}\|\mathbf{Z}_{\mathbf{\Theta}_{M.e}} - \mathbf{Z}_{\mathbf{\Theta}_{M.e}}\mathbf{\Theta}_s\|_F^2 + \frac{\lambda_2}{2}\sum_{m=1}^{M}\|\mathbf{X}^m - \hat{\mathbf{X}}_{\mathbf{\Theta}}^m\|$$
$$\text{s.t } \text{diag}(\mathbf{\Theta}_s) = \mathbf{0}, \tag{4.6}$$

where $\mathbf{\Theta}$ denotes all the training network parameters including $\mathbf{\Theta}_{M.e}$, $\mathbf{\Theta}_s$ and $\mathbf{\Theta}_{M.d}$. The joint representation is denoted by $\mathbf{Z}_{\mathbf{\Theta}_{M.e}}$, and $\hat{\mathbf{X}}_{\mathbf{\Theta}}^m$ 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.

4. 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 4.4) 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

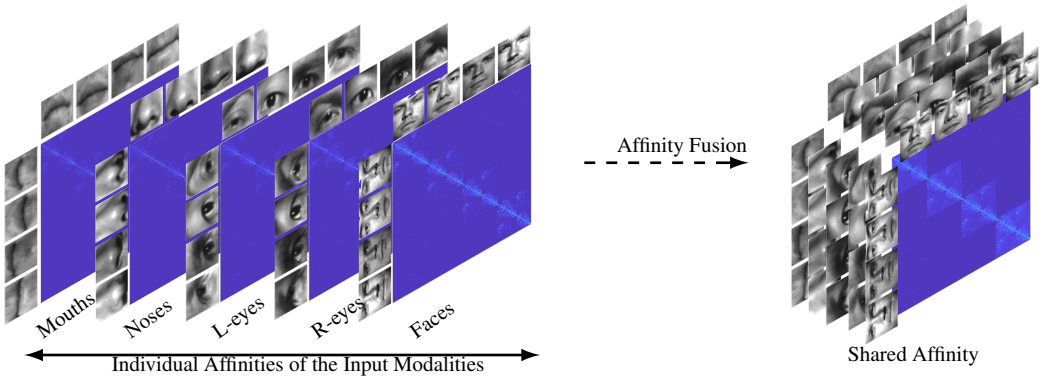


Figure 4.5: An example of affinity fusion. Affinities corresponding to different modalities are combined to have only a single shared affinity. This method does not relay on spatial relation across different modalities. Instead, it aggregates the similarities among data points across different modalities and returns a shared affinity.

similar (dissimilar) data in one modality should be similar (dissimilar) in the other modalities as well. Figure 4.5 shows an example of the proposed affinity fusion method by forcing the modalities to share the same affinity matrix.

In the DSC framework [4], the affinity matrix is calculated from the self-expressive layer weights as follows

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

where $\mathbf{\Theta}_s$ corresponds to the self-expressive layer weights learned by an end-to-end training strategy [4]. Thus a shared $\mathbf{\Theta}_s$ results in a common $\mathbf{W}$ across the modalities. We enforce the modalities to share a common $\mathbf{\Theta}_s$ while having different encoders, decoders and the latent representations.

4..1 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

Algorithm 2 Spatial and affinity fusion algorithms

procedure DMSC($\{\mathbf{X}^m\}_{m=1}^M, \lambda_1, \lambda_2,$ `'mode'`)
  if `mode` = Spatial fusion **then**
    Train the networks using the loss (4.6).
  else if `mode` = Affinity fusion **then**
    Train the networks using the loss (4.7).
  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.

across all the M modalities. Figure 4.3(d) gives an overview of the proposed affinity fusion-based network architecture.

4..2 End-to-End Training

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

$$\min_{\boldsymbol{\Theta}} \ \|\boldsymbol{\Theta}_s\|_p + \frac{\lambda_1}{2}\sum_{m=1}^{M}\|\mathbf{Z}^m_{\boldsymbol{\Theta}^m_e} - \mathbf{Z}^m_{\boldsymbol{\Theta}^m_e}\boldsymbol{\Theta}_s\|_F^2 + \frac{\lambda_2}{2}\sum_{m=1}^{M}\|\mathbf{X}^m - \hat{\mathbf{X}}^m_{\boldsymbol{\Theta}^m}\| \text{ s.t. } \operatorname{diag}(\boldsymbol{\Theta}_s) = \mathbf{0}, \tag{4.7}$$

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 (4.7) encourages the networks to learn the latent representations that share the same affinity matrix.

Algorithm 2 summarizes the proposed spatial fusion and affinity fusion-based subspace clustering methods.

Experiment	Dataset	# ofmodalities	# of samplesper modality
Digits	MNIST [5], USPS [6]	2	2000
Heterogeneous Faces	ARL [7]	5	2160
Facial components	Extended Yale-B [8]	5	2432

Table 4.1: 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.

5. 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 [5] and the USPS [6] handwritten digits datasets. Here, we view an image 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 [7]. 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 [8]. We extract facial components (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 8.5 (a), (b), and (c) show sample images from the digits, ARL and Extended Yale-B datasets, respectively. Table 4.1 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 [4], 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,

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.

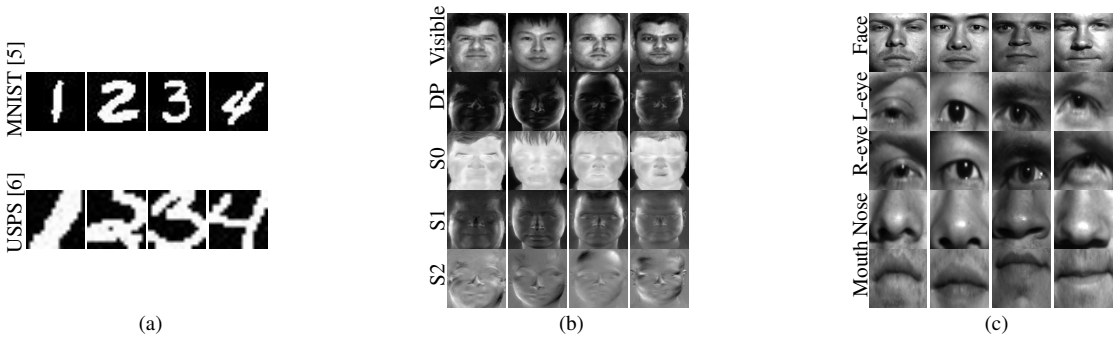
Table 4.2: Spatial fusion variations that are used in the experiments.

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 4.2 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 [75], TM-MSC [73], MSSC [49], MLRR [49], KMSSC [49], and KMLRR [49].

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 [36] and LRR [37] performances on the individual modalities along with the recently proposed DSC method [4]. Furthermore, we train an encoder-decoder similar to the network in [4] 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.

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

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



(a) (b) (c)

Figure 4.6: Sample images from (a) MNIST [5], and USPS [6] digits datasets, (b) ARL polarimetric face dataset [7], and (c) Faces and facial components from the Extended Yale B dataset [8]. In our experiments, samples from all the modalities are resized to 32×32, and rescaled to have pixel values between 0 and 255.

		DSC[4]	AE+SSC	SSC[36]	LRR[37]
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 4.3: 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.

are stacked three deconvolution layers mimicking the inverse task of the encoder.

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 [4] with unimodal experiments.

Training details: We implemented our method in Python-2 with Tensorflow-1.4 [128]. We use the adaptive momentum-based gradient descent method (ADAM) [129] 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 (4.2), (4.6) and (4.7) while training the networks. Similar to [4], 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 [4] as well. A sensibility analysis over the range $[10^{-4}, 10^4]$ in Section 5..5, 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) [130], and Adjusted Rand Index (ARI) [131] 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].

5..1 Handwritten Digits

In the first set of experiments, we use the 10 classes (i.e. digits) from the MNIST and the USPS datasets. Figure 8.5 (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 8.5 (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.

Note that some structures including the late fusion methods in Table 4.2 and the affinity fusion method have more than one branches in some of their layers.

Table 4.3 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 [132].

Note that while the DSC method in Table 4.3 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 4.4 compares the performance of the multimodal methods in terms of both clustering error rates and NMI. We observe that most of the multimodal methods can successfully integrate the complementary information of the

		CMVFC[75]	TM-MSC[73]	MSSC[49]	MLRR[49]	KMSSC[49]	KMLRR[49]	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 4.4: 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.

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 error rate of only 4.85%, and an NMI metric of above 90%.

5..2 ARL Heterogeneous Face Dataset

To test our methods on clustering datasets with a large number of subjects, we use the ARL dataset [7] 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 8.5 (b).

Table 4.5 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

		DSC[4]	AE+SSC	SSC[36]	LRR[37]
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 4.5: 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.

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 4.4. 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

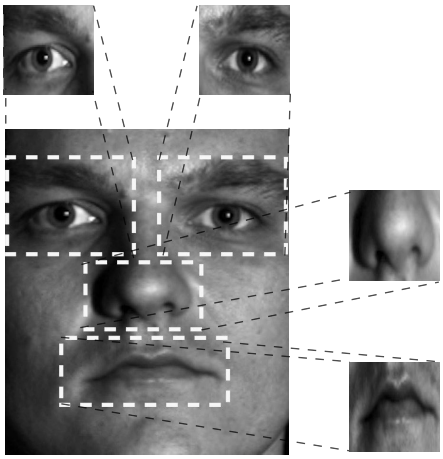
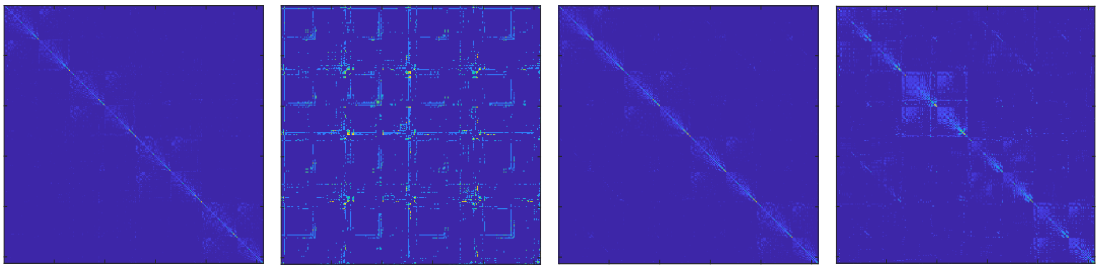


Figure 4.7: Facial components are extracted by applying a fixed mask on the faces in the Extended Yale B dataset [8].



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

Figure 4.8: 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.

dataset.

5..3 Facial Components

The Extended Yale B dataset [8] consists of 64 frontal images of 38 individuals under varying illumination conditions. This dataset is popular in subspace clustering studies [4, 37, 36]. 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 [111]. To crop the facial components, we apply a fixed face mask as shown in Figure 4.7 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

		DSC[4]	AE+SSC	SSC[36]	LRR[37]
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 4.6: 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.

mouth modality cannot be projected on the spatial positions in the nose modality. Sample images from this dataset are shown in Figure 8.5 (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 4.6. 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 4.4. It is worth highlighting several interesting observations from the results. As can be seen, the max-pooling fusion function in the *late-mpool* and *interm-mpool* methods fails 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 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 below 1% error rate and the NMI metric of 98.89%. This is mainly due to the fact that this method does not rely on the spatial correspondence among the modalities.

Figure 4.8 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 4.8 (a) shows the affinity matrix corresponding to the DSC method for clustering faces. Figure 4.8 (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 4.8 (c) and Figure 4.8 (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.

5..4 Convergence study

To empirically show the convergence of our proposed method, in Figure 4.9, we show the objective function of the *affinity fusion* method and its clustering metrics vs iteration plot for solving (4.7). The reported values in Figure 4.9 are normalized between zero and one. As can be seen from the figure, our algorithm converges in a few iterations.

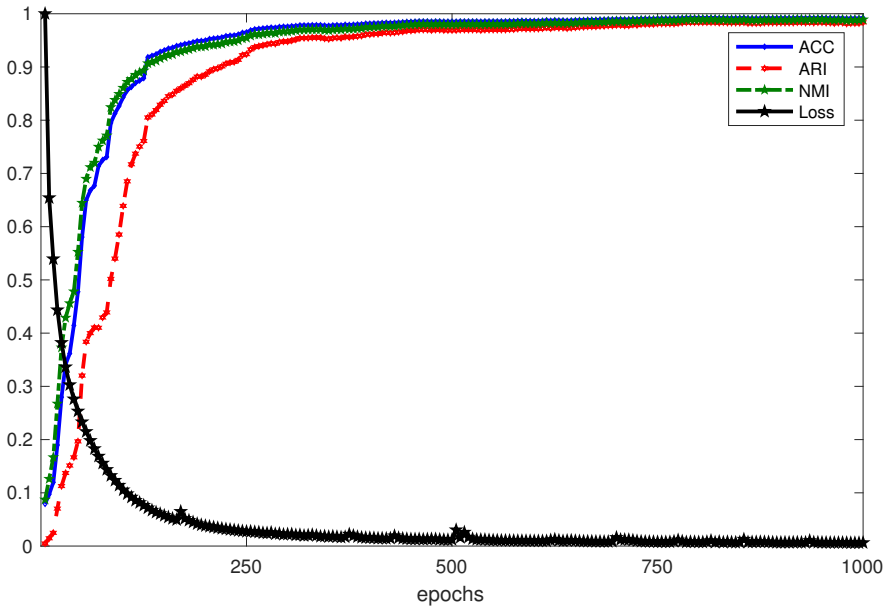


Figure 4.9: The *affinity fusion* method's loss function and the clustering metrics over different training epochs in the Yale B facial components experiment. The reported values in this figure are normalized between zero and one. This figure shows the convergence of our objective function.

5..5 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 (4.7). Figure 4.10 shows the influence of these regularization parameters on the performance of the *affinity fusion* method on the Extended Yale-B dataset.

In Figure 4.10 (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 4.10 (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 [4] also performed well in all the experiments.

5..6 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 (4.7). Table 4.7 reports the clustering metrics for the experiments with $p = 0.3$, $p = 1$, $p = 1.5$ and $p = 2$. As can be

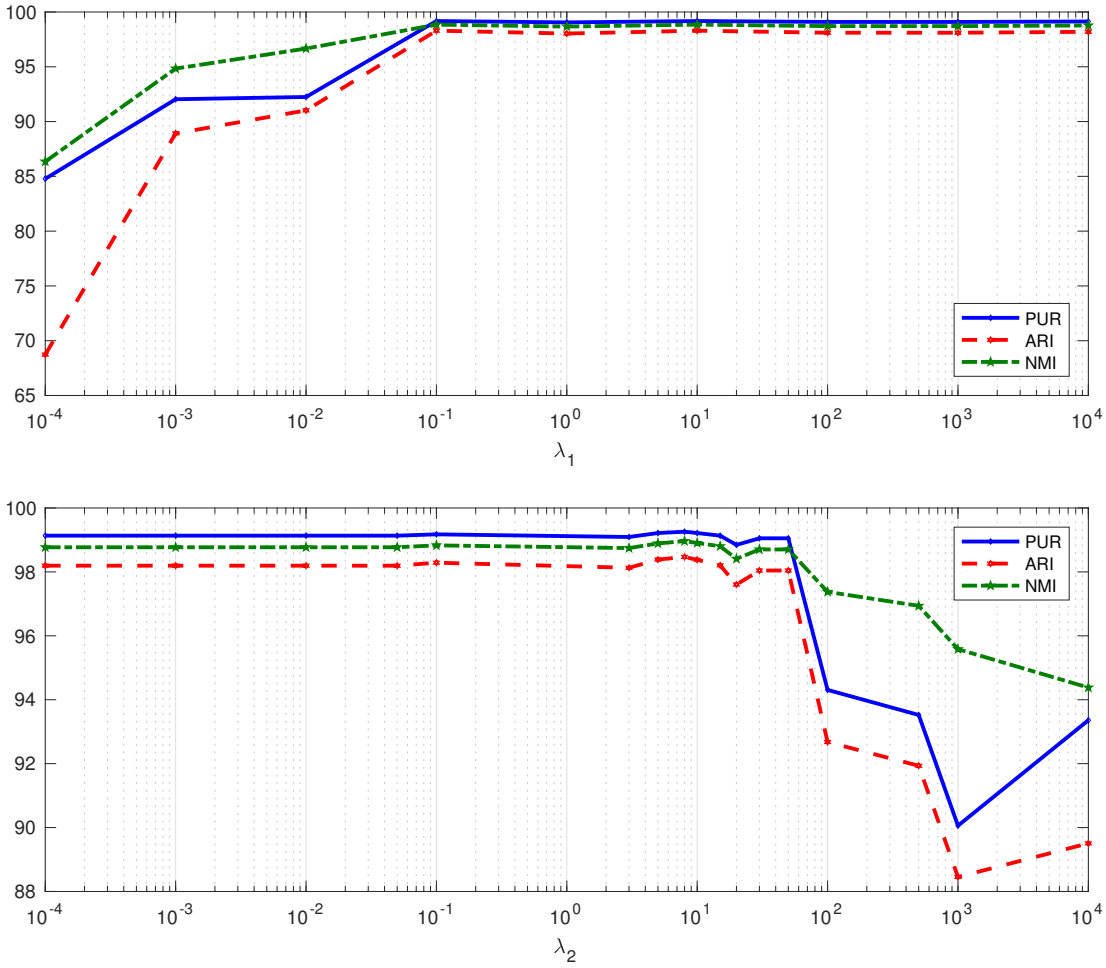


Figure 4.10: The *affinity fusion* method's performance through different parameter selections for λ_1 and λ_2.

seen from this table, while experiments with $p = 1$, $p = 1.5$ and $p = 1$ 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 (4.7) 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.

6. 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

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 4.7: 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.

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 error rate of 0.78% and normalized mutual information of 98.89%.

Chapter 5

Deep Subspace Clustering with Data Augmentation

1. Introduction

Recent advances in technology have provided massive amounts of complex high-dimensional data for computer vision and machine learning applications. High-dimensionality has adverse effects, including confusion of algorithms with irrelevant dimensions and *curse of dimensionality* as well as increased computation time and memory [19, 133]. This motivates us to explore techniques for representing high-dimensional data in lower dimensions. In many practical applications such as face images under various illumination conditions [16] and hand-written digits [17], high-dimensional data can be represented by union of low-dimensional subspaces. The subspace clustering problem aims at finding these subspaces. In particular, the objective of subspace clustering is to find the number of subspaces, their basis and dimensions, and assign data to these subspaces [19].

Conventional subspace clustering algorithms assume that data lie in linear subspaces [36, 34, 35, 39, 134]. In practice, however, many datasets are better modeled by non-linear manifolds. To deal with this issue, many works have incorporated projections and kernel tricks to express non-linearity [108, 107, 39, 135, 136, 137]. Recently, deep subspace clustering (DSC) methods [64, 4, 65, 66, 51] have been proposed which essentially learn unsupervised nonlinear mappings by projecting data into a latent space in which data lie in linear subspaces. Deep subspace clustering networks have shown promising performances on various datasets.

Deep learning techniques are prone to overfitting. Data augmentation is often presented as a type of regularization to mitigate this issue [138, 139]. While data augmentation for deep learning-based methods have proven to be beneficial, the current framework of DSC networks is unable to take the full advantage of data augmentation. In this work, we modify the DSC framework and propose a model that can incorporate data augmentation into DSC.

An important difference between data augmentation in subspace clustering and data augmentation in supervised tasks is the fact that as opposed to supervised tasks, we do not have ground-truth labels for the existing samples in the subspace clustering algorithms. Corresponding to the fact that objects remain the same even if we slightly transform them, in supervised deep learning models, transformations of an existing sample are trained to be predicted with a consistent label similar to the ground-truth label of the original sample. How can one convey such property in an unsupervised subspace clustering task, where the data does not have the ground-truth labels?

A DSC model should favor functions that give consistent outputs for similar data points with a slight difference in their percept. To achieve this, we optimize a consistency loss that is based on temporal ensembling. We input plausible transformations of existing samples into the model and require the autoencoders of the model to map the transformations to consistent subspaces similar to the subspace of the original data.

Efficient augmentation policies improve the performance of the deep networks. However, not all the image transformations construct efficient augmentation policies. Efficient augmentation policies can be different from a dataset to another [140, 141, 141]. In supervised applications, the validation set is often used to manually search among transformations such as rotation, horizontal flip, or translation by a few pixels to find efficient augmentations. Manual augmentation needs prior knowledge and expertise, and it can only search among a handful of predefined trials. Some methods automate this search for classification networks [140, 142, 143]. However, these methods are only designed for the classification task and cannot be applied to the task of subspace clustering. This is because we do not have a validation or training set in subspace clustering. We overcome this issue by providing a simple yet effective method for finding efficient augmentation policies using a greedy search and use mean Silhouette scores to evaluate the effect of different augmentation policies on the performance of our proposed model.

2. Related Work

Clustering Methods with Augmentation. A recent method proposes a technique for deep embedded clustering algorithms with augmentations [144, 145]. In the pre-training stage they use augmentations in training autoencoders, and in the fine-tuning stage they encourage the augmented data to have the same centroid as their corresponding data. To the best of our knowledge, we are the first to propose an augmentation framework for deep subspace clustering algorithms.

Self-supervision with Consistency Loss. The idea of learning consistent features for different transformations of unlabeled data has been used in a number of works largely in the semi-supervised and self-supervised learning literature [146, 147, 148, 149, 150, 151].

Self-expressiveness Models in 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 $\mathcal{S}_1 \cup \mathcal{S}_2 \cup \cdots \cup \mathcal{S}_n$. 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 $N_1 + N_2 + \cdots + N_n = N$.

Due to their simplicity, theoretical correctness, and empirical success, subspace clustering methods that are based on *self-expressiveness property* are very popular [63]. Self-expressiveness property can be stated as

$$\mathbf{X} = \mathbf{X}\mathbf{C} \quad s.t \quad \text{diag}(\mathbf{C}) = \mathbf{0}, \tag{5.1}$$

where $\mathbf{C} \in \mathbb{R}^{N \times N}$ is the coefficient matrix. There may exist many coefficient matrices that satisfy the condition in (5.1). Among those, *subspace preserving* solutions are especially of interest to self-expressiveness based subspace clustering methods. Subspace preserving property states that if an element in $\mathbf{C}$ is non-zero, the two data points in $\mathbf{X}$ that correspond to this coefficient are in the same subspace.

Self-expressiveness based methods combine these two properties and solve a problem of the form:

$$\min_{\mathbf{C}} \mathcal{L}_{\text{S.E.}}(\mathbf{C}, \mathbf{X}) + \lambda_1 \mathcal{L}_{\text{S.P.}}(\mathbf{C}), \tag{5.2}$$

where λ_1 is a regularization constant, $\mathcal{L}_{\text{S.E.}}$ and $\mathcal{L}_{\text{S.P.}}$ impose the self-expressiveness and subspace-preserving properties, respectively. Most of the linear methods use $\mathcal{L}_{\text{S.E.}}(\mathbf{C}, \mathbf{X}) = \|\mathbf{X} - \mathbf{X}\mathbf{C}\|_F^2$.

However, for $\mathcal{L}_{\text{S.P.}}(\mathbf{C})$, different methods use various regularizations, including ℓ_1-norm, ℓ_2-norm and nuclear norm [36, 63, 37].

In recent years, deep neural network-based extensions were introduced to self-expressiveness based models [64, 4, 65, 66]. For these methods, x_is do not need to be drawn from a union of linear subspaces. Instead, they use autoencoder networks to map the data points to a latent space where data points lie into a union of linear subspaces and exploit the self-expressiveness and subspace-preserving properties in the latent space. Let $\mathbf{Z} \in \mathbb{R}^{d \times N}$ be the latent space features developed by the encoder in the autoencoders. Deep subspace clustering networks solve a problem of the form:

$$\min_{\Theta} \mathcal{L}_{\text{S.E.}}(\mathbf{C}, \mathbf{Z}) + \lambda_1 \mathcal{L}_{\text{S.P.}}(\mathbf{C}) + \lambda_2 \mathcal{L}_{\text{Rec.}}(\mathbf{X}, \hat{\mathbf{X}}), \tag{5.3}$$

where λ_1 and λ_2 are regularization constants, Θ is the union of trainable parameters, $\hat{\mathbf{X}}$ is the reconstruction of $\mathbf{X}$ and the output of the decoder, and $\mathcal{L}_{\text{Rec.}}(\mathbf{X}, \hat{\mathbf{X}}) = \|\mathbf{X} - \hat{\mathbf{X}}\|_F^2$ is the reconstruction loss in training the autoencoder. Once a proper $\mathbf{C}$ is found from (5.2) or (5.3), spectral clustering methods [67] are applied to the affinity matrix $\mathbf{W} = |\mathbf{C}| + |\mathbf{C}|^T$ to obtain the segmentation of the data $\mathbf{X}$.

3. Deep Subspace Clustering Networks with Data Augmentation

The human brain considers an object to remain the same, even if the percept changes slightly. Correspondingly, when data augmentation is used in supervised deep learning models, transformations of existing samples are trained to predict consistent labels similar to the ground-truth label of original samples. Conveying the same insight, we argue that a DSC model should favor functions that give consistent outputs for similar data points. We approach this property by keeping the estimated subspace membership of data points consistent when an augmentation policy is applied to them. During the training process, we smooth the predictions for the subspace memberships via temporal ensembling of estimated affinity matrices from previous iterations.

Let $\mathbf{X}^t = [\mathbf{x}_1^t, \cdots, \mathbf{x}_N^t] \in \mathbb{R}^{D \times N}$ be the transformed version of N existing data points $\mathbf{X} = [\mathbf{x}_1, \cdots, \mathbf{x}_N] \in \mathbb{R}^{D \times N}$ at the iteration t. $\mathbf{X}^t$ is the observation at time t when an augmentation policy is applied to the existing data points $\mathbf{X}$.

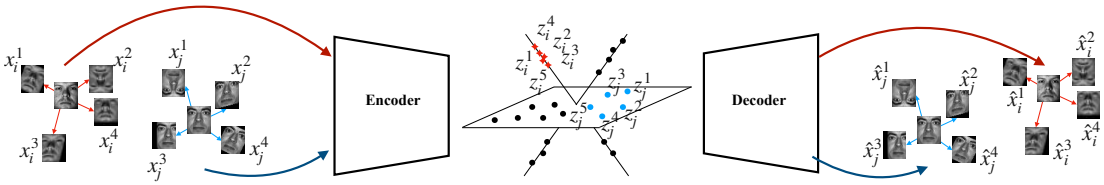


Figure 5.1: An overview of the proposed deep subspace clustering networks with data augmentation. The existing data points x_i and x_j are transformed into x_i^t and x_j^t in each iteration by an augmentation policy. However, the autoencoder learns to keep their latent space features within consistent subspaces.

Our model can be applied to a variety of DSC networks. In this section, we consider a general form that consists of an encoder that takes $\mathbf{X}^t$ as an input and generates latent space features $\mathbf{Z}^t$. The latent space features are reconstructed by a self-expressive layer with parameters $\mathbf{C}^t$. That is, $\mathbf{Z}^t\mathbf{C}^t$ is fed to the decoder to develop $\hat{\mathbf{X}}^t$, which is a reconstruction of $\mathbf{X}^t$. Figure 5.1 shows an overview of this model. Note that such a model includes a fully-connected layer that connects all the samples in the mini-batch (the self-expressive layer). Thus, the number of data points and their orders cannot be changed during the training. We keep a placeholder with N fields that correspond to the existing samples and feed $\mathbf{X}^t$ to this placeholder at every training step t. The permutation of samples in $\mathbf{X}^t$ remains the same.

As mentioned, we aim for an autoencoder that preserves the subspace membership of slightly transformed inputs. Let $\mathbf{C}^t$ be the coefficient matrix that is constructed at the t-th iteration of a subspace clustering algorithm. In addition, let $\hat{\mathbf{Q}}$ be an existing estimation of subspace membership matrix, whose rows are one-hot vectors denoting the subspace memberships assigned to different samples. The multiplication of $\hat{\mathbf{Q}}^T$ and $|\mathbf{C}^t|$ gives a matrix whose (i,j)th element shows the contribution of the samples assigned to the i-th subspace in reconstructing the j-th sample. For a perfect subspace-preserving coefficient matrix, $\hat{\mathbf{Q}}^T|\mathbf{C}^t|$ has only one non-zero element in each row.

For each sample j, the maximum value in the j-th row of $\hat{\mathbf{Q}}^T|\mathbf{C}^t|$ can point to a new estimate for its subspace membership. Therefore, a prediction of subspace membership matrix at the iteration t can be calculated as follows

$$\mathbf{Q}^t = \text{Softmax}(\hat{\mathbf{Q}}^T|\mathbf{C}^t|), \tag{5.4}$$

where Softmax$(\cdot)$ corresponds to the softmax function on the rows of its input. We refer to $\mathbf{Q}^t$

as *temporal* subspace membership matrix.

The temporal subspace membership matrix $\mathbf{Q}^t$ estimates the subspace memberships for the current observation $\mathbf{X}^t$. Note that because of the randomly augmented inputs, the coefficient matrix $\mathbf{C}^t$ can undergo sudden changes in different time frames. While it is fine to have different coefficient matrices for slight transformations of data, we are interested in maintaining persistent subspace membership matrices $\mathbf{Q}^t$. Thus, we propose a subspace membership consistency loss.

We keep an exponential moving average (EMA) of $\mathbf{C}^t$s, the coefficient matrices, to provide a smooth temporal ensemble for the coefficient matrix. Thus, in addition to the *temporal* subspace membership matrix in (5.4), in each training iteration, we can calculate another membership matrix corresponding to the temporal ensemble of coefficient matrices in prior iterations. We refer to this membership matrix as $\mathbf{Q}^t_{\text{Ens.}}$.

Let $\mathbf{C}^{t-1}_{\text{EMA}}$ be the EMA of coefficient matrices until the iteration $t-1$, and $\mathbf{C}^t$ be the calculated update for the coefficient matrix at the iteration t. The EMA of the coefficient matrix at the iteration t can be updated as follows

$$\mathbf{C}^t_{\text{EMA}} = \alpha \mathbf{C}^{t-1}_{\text{EMA}} + (1-\alpha)\mathbf{C}^t, \tag{5.5}$$

where $0 < \alpha < 1$ is the smoothing factor. Using $\mathbf{C}^t_{\text{EMA}}$ we can calculate $\mathbf{Q}^t_{\text{Ens.}}$ as

$$\mathbf{Q}^t_{\text{Ens.}} = \text{Softmax}(\hat{\mathbf{Q}}^T |\mathbf{C}^t_{\text{EMA}}|), \tag{5.6}$$

where $\hat{\mathbf{Q}}$ is the same prior membership matrix as in (5.4).

Note that $\mathbf{Q}^t_{\text{Ens.}}$ provides more consistent subspace membership predictions as compared to $\mathbf{Q}^t$. To encourage the autoencoders to favor functions that preserve the subspace memberships even for differently transformed observations $\mathbf{X}^t$, we propose the subspace membership consistency loss as follows:

$$\mathcal{L}_{\text{Cons.}}(\mathbf{Q}^t_{\text{Ens.}}, \mathbf{Q}^t) = \text{CE}(\mathbf{Q}^t_{\text{Ens.}}, \mathbf{Q}^t), \tag{5.7}$$

where $\text{CE}(\cdot)$ denotes the cross-entropy function. $\mathcal{L}_{\text{Cons.}}$ penalizes the temporal changes to the subspace memberships if they are inconsistent with the temporal ensemble of subspace memberships $\mathbf{Q}^t_{\text{Ens.}}$.

Full Objective. We train the networks iteratively with two steps of subspace clustering and subspace membership consistency in each iteration. In the subspace clustering step, the loss function of the subspace clustering algorithm of choice (5.3) is optimized, and in the subspace membership consistency step, (5.7) is optimized. That is at each iteration t, we train the netwrok with the following algorithm.

$$\begin{cases} \text{Step 1:} & \min_{\Theta}(\mathcal{L}_{\text{S.E.}}(\mathbf{C}^t, \mathbf{Z}^t) + \lambda_1 \mathcal{L}_{\text{S.P.}}(\mathbf{C}^t) + \lambda_2 \mathcal{L}_{\text{Rec.}}(\mathbf{X}^t, \hat{\mathbf{X}}^{\mathbf{t}})), \\ \text{Step 2:} & \min_{\Theta}(\mathcal{L}_{\text{Cons.}}(\mathbf{Q}^t_{\text{Ens.}}, \mathbf{Q}^t)), \end{cases} \tag{5.8}$$

where Θ is the union of trainable parameters in the networks.

4. Finding Efficient Augmentations

In the previous section, we denoted $\mathbf{X}^t$ as a stochastic transition of $\mathbf{X}$ which is the result of applying an augmentation policy. The choice of augmentation policy plays an important role in the performance of the network. We formulate the problem of finding the best augmentation policy as a discrete search problem.

Our method consists of three components: A *score*, a *search algorithm* and a *search space* with n_s possible configurations. The search algorithm samples a data augmentation policy S_i, which has information about what image processing operation to use, the probability of using the operation in each iteration, and the magnitude of the operation. The policy S_i will be used to train a child deep subspace clustering network with a fixed architecture. The trained child network will return a score that specifies the effect of applying the policy S_i to the input data on the performance of deep subspace clustering task. Finally, all the tested policies $\{S_i\}_1^{n_s}$ will be sorted based on the returned scores.

In the following, we describe the *score* , the *search algorithm* and the *search space* in detail.

Score. In our framework, the score is a metric that evaluates the performance of the DSC on a certain given input. Note that the ground-truth labels are unknown at this stage. Therefore, we need to use a validation technique that does not use the ground-truth labels. Any internal validation of clustering methods [152, 153, 154] , including mean Silhouette coefficient [152] or the Davies-Bouldin index (DBI) [153] can serve as the score metric in our search. We use mean Silhouette coefficient in this chapter.

Search Space. In our search space, a sample policy S_i consists of ℓ sequential sub-policies with each sub-policy using an image operation. Additionally, each operation is also associated with two hyper-parameters: 1) the probability of applying the operation, and 2) the magnitude of the operation. We discretize the range of probability and magnitude values into n_p and n_m discrete values, respectively (with uniform spacing). This way, we can use a discrete search algorithm to find them. For n_o operations, this constructs a search space with the size of $n_s = (n_o \times n_p \times n_m)^\ell$.

Search Algorithm. The size of search space n_s, can grow exponentially. A brute-force search might be impractical. To make the searching process feasible, we use a greedy search []. First, we begin searching in the reduced search space where each sample policy has only one sub-policy ($\ell = 1$). In the reduced search space, we find the best probability and magnitude for each image operation. Note that n_p and n_m can also be decreased as much as necessary to keep the search tractable.

Once we find the best augmentation operations for the first sub-policy, we search for the second sub-policy ($\ell = 2$). For each found sub-policy in the first stage, we search for the best combination of image operations and their probabilities and magnitudes.

This process continues until we reach $\ell = \ell_{\max}$, the maximum number of sub-policies. At this point, we sort all the potentially good policies that are found until this point, and select the best b augmentation policies among them.

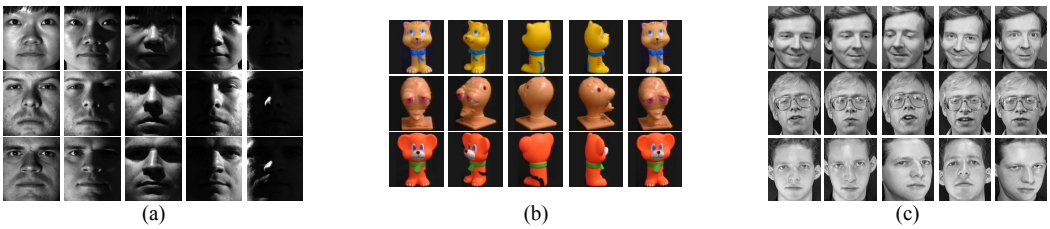
5. Experimental Results

We evaluate our method against state-of-the-art subspace clustering algorithms on three standard datasets. We first use the algorithm described in section 4. to find the best augmentation policies for each dataset. Then, we use the found augmentation policies in the ablation study as well as in comparisons with state-of-the-art subspace clustering algorithms.

We use the following datasets in our experiments:

Extended Yale-B dataset [8] contains 2432 facial images of 38 individuals from 9 poses and under 64 illuminations settings.

ORL dataset [11] includes 400 facial images from 40 individuals. This corresponds to only 10 samples per subject.



(a) (b) (c)

Figure 5.2: Sample images from different used datasets. (a) Extended Yale-B dataset [8]. (b) COIL dataset [9, 10] . (c) ORL dataset [11].

COIL-100 [9] and **COIL-20** [10] datasets are respectively consisted from images of 100 and 20 objects placed on a motorized turnable. Per each object, 72 images are taken at pose intervals of 5 degrees that covers a 360 degrees range. Following most of the prior studies, in our experiments, we use grayscale images of these datasets.

Figure 8.5 (a), (b), and (c) show sample images from the Extended Yale-B, ORL and COIL datasets, respectively. Note that in the subspace clustering tasks, the datasets are not split into training and testing sets. Instead, all the existing samples are used in both the learning stage and the performance evaluation stage.

Experimental Setups. While our method can be applied to many DSC algorithms, unless otherwise stated, due to its promising performance, we adopt the MLRDSC networks [66] and apply our method to its networks. We call the result MLRDSC with Data Augmentation (MLRDSC-DA). The objective function of MLRDSC can be also written in the format of (5.3). The *self-expressiveness* and *subspace-preserving* loss terms in MLRDSC are

$$\mathcal{L}_{\text{S.E.}}(\mathbf{C},\mathbf{Z}) = \sum_{l=1}^{L} \|\mathbf{Z}_l - \mathbf{Z}_l(\mathbf{G}+\mathbf{D}_l)\|_F^2 \quad \text{and} \quad \mathcal{L}_{\text{S.P.}}(\mathbf{C}) = \|\mathbf{Q}^T|\mathbf{G}|\|_1 + \lambda_3 \sum_{l=1}^{L} \|\mathbf{D}_l\|_F^2, \tag{5.9}$$

where L is the number of layers in the autoencoder, $\mathbf{Z}^l$ is the features at the l-th layer, and $\mathbf{C} = \mathbf{G} + \frac{1}{L}\sum_{l=1}^{L}\mathbf{D}_l$. The coefficient matrix in this model is calculated by the consistency matrix $\mathbf{G}$ and distinctive matrices $\{\mathbf{D}_l\}_{l=1}^{L}$. The distinctive matrices enforce subspace-preserving across different layers, and $\mathbf{G}$ captures the shared information between the layers.

In the training of MLRDSC-DA, we first pre-train the networks by performing the ML-RDSC algorithm. Then, we continue training MLRDSC-DA for a few additional iterations

until convergence with (5.8) as

$$\begin{cases} \text{Step 1:} \quad \min_{\Theta} \quad \sum_{l=1}^{L} \|\mathbf{Z}_l^t - \mathbf{Z}_l^t(\mathbf{G}^t + \mathbf{D}_l^t)\|_F^2 + \lambda_1 \|\mathbf{Q}^{t^T} |\mathbf{G}^t|\|_1 \\ \qquad\qquad\qquad\qquad +\lambda_3 \sum_{l=1}^{L} \|\mathbf{D}_l^t\|_F^2 + \lambda_2 \|\mathbf{X}^t - \hat{\mathbf{X}}^{\mathbf{t}}\|_F^2, \\ \text{Step 2:} \quad \min_{\Theta} \quad \text{CE}(\mathbf{Q}_{\text{Ens.}}^t, \mathbf{Q}^t), \end{cases} \tag{5.10}$$

where we shape the temporal coefficient matrix as $\mathbf{C}^t = \mathbf{G}^t + \frac{1}{L}\sum_{l=1}^{L} \mathbf{D}_l^t$, and $\mathbf{Q}_{\text{Ens.}}^t$ and $\mathbf{Q}^t$ are calculated from (5.6) and (5.4), respectively.

We use the same training settings as described in [66]. This includes the same architecture for networks and values for the hyper-parameters λ_1, λ_2, λ_3 in different experiments as well as the initial values of a zero matrix for the membership matrix $\hat{\mathbf{Q}}$, and matrices with all the elements equal to 0.0001 for the coefficient matrices $\mathbf{G}^0$ and $\mathbf{D}_l^0$s at the iteration $t = 0$. We update $\hat{\mathbf{Q}}$ every 50 iterations by substituting the subspace membership estimations with the result of subspace clustering performed on the current $\mathbf{C}^t$. We set the EMA decay to $\alpha = 0.999$ in all the experiments (selected by cross-validation and mean silhouette coefficient as the evaluation metric). We implemented our method with PyTorch. We use the adaptive momentum-based gradient descent method (ADAM) [129] with a learning rate of 10^{-3} to minimize the loss functions. Similar to other DSC methods, we input the whole dataset as a batch. In all the conducted experiments, we report 5-fold averages.

5..1 Best Augmentation Policies Found on the Datasets

We perform the search algorithm in Section 4. on different datasets to find the best augmentation policies for each dataset. To reduce the computations, we search in the search space of augmentation policies with the maximum number of sub-policies $\ell_{\text{max}} = 2$ (i.e. up to two sub-policies can be combined to construct a policy), and set the probability to $p = 0.1$ and the magnitude to $m = 0.3 \times r$ where $r = (\text{max}-\text{min})$ is the magnitude range that image operations accept. The image operation search space is the following set: {FlipLR, ShearX, FlipUD, SearY, Posterize, Rotate, Invert, Brightness, Equalize, Solarize, Contrast, TranslateY, TranslateX, AutoContrust, Sharpness, Cutout} that is also used in [140]. This results in a search space of $n_s = 16^2$. We selected the values for magnitude and probability of augmentation polices by searching in the full search space of augmentation policies for the first two subjects in the Extended Yale B dataset.

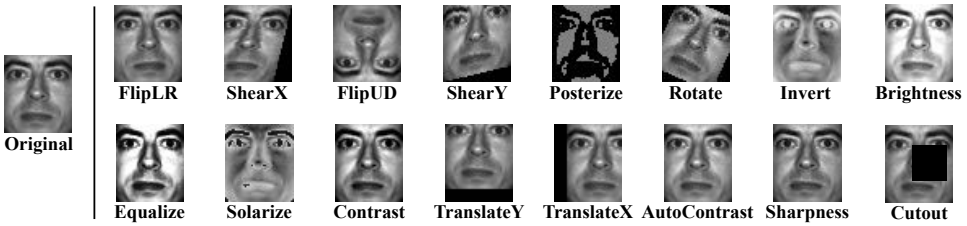


Figure 5.3: Different image transformations on a sample from the Extended Yale B dataset.

Table 5.1: Augmentation policies that yield the highest mean Silhouette coefficient in the subspace clustering results on different datasets.

Dataset	Augmentation Policy 1	Augmentation Policy 2
Extended Yale B	(Op =‘`ShearY`’, m=0.3r, p=0.1)	(Op =‘`TranslateY`’, m=0.3r, p=0.1) + (Op =‘`Contrast`’, m=0.3r, p=0.1)
COIL-20 & COIL-100	(Op =‘`Posterize`’, m=0.3r, p=0.1) + (Op =‘`Sharpness`’, m=0.3r, p=0.1)	(Op =‘`FlipLR`’, p=0.1) + (Op =‘`Contrast`’, m=0.3r, p=0.1)
ORL	(Op =‘`ShearX`’, m=0.3r, p=0.1) + (Op =‘`Sharpness`’, m=0.3r, p=0.1)	(Op =‘`FlipLR`’, p=0.1) + (Op =‘`ShearX`’, m=0.3r, p=0.1)

Figure 5.3 shows the different augmentation policies applied to a sample drawn from the Extended Yale B dataset. The details of these image operations are described in Table 1 in the supplementary materials.

For each candidate augmentation policy, we train our MLRDSC-DA model, perform subspace clustering, and return the mean Silhouette coefficient [152] as the clustering performance. We use the mean Silhouette coefficients to sort the augmentation policies(including policies with $\ell < \ell_{max}$ sub-policies) and select the top two performing augmentation policies in each dataset. That is $b = 2$.

Table 5.1 shows the found augmentation policies that yield to the highest Silhouette coefficients in the subspace clustering results on different datasets. In our experiments, COIL-20 and COIL-100 resulted in similar policies. Unless otherwise stated, in all the experiments, we apply these augmentation policies to the inputs of our MLRDSC-DA algorithm.

5..2 Ablation Study and Analysis of The Model

To understand the effects of some of our model choices, we explore some ablations of our model on the Extended Yale B dataset. In particular, we test our model on two different deep subspace clustering methods, DSC [4] and MLRDSC [66], and in four settings where 1) the

Table 5.2: Ablation study of our method in terms of clustering error (%) on Extended Yale B. Top performers are bolded.

Backbone	Augmentations × Consistency Loss ×	Augmentations ✓ Consistency Loss ×	Augmentations × Consistency Loss ✓	Augmentations ✓ Consistency Loss ✓
DSC	2.67	3.10	2.56	**1.92**
MLRDSC	1.36	2.84	0.95	**0.82**

consistency loss exists or 2) is ablated; 3) the optimal augmentations policies are applied to the inputs or 4) the data is fed without any augmentations.

If we remove both augmentations and the consistency loss, our networks, based on their backbones, turn to either DSC or MLRDSC networks. In the versions that data augmentation is applicable, the augmentations in Table 5.1 are used. Further analysis on the evaluation of the found augmentation policies is provided in section 5..4.

We report the performances in Table 8.7. As can be seen, the top performer is our full model with augmentations and the consistency loss applied to the MLRDSC method. MLRDSC-based methods, in general, outperform DSC-based methods. Consistency loss slightly improves the performance even without data augmentation. This is the result of temporal ensembling.

As can be seen in the second column of this table, applying the found augmentations to the input of DSC and MLRDSC networks without further modification (i.e., not adding the consistency loss) not only does not improve the results, but it slightly degrades the performance. These results clearly show both the importance of the consistency loss and the benefit of using data augmentations when it is combined with the consistency loss.

5..3 Comparison with State-of-The-Art Subspace Clustering Methods

In this section, we evaluate our method against the state of the art subspace clustering methods. We apply the found augmentation policies in Table 5.1 to the data on Extended Yale B, ORL, COIL-20 and COIL-100 datasets and feed them to our MLRDSC-DA method.

The rows in Table 5.4 report the clustering error rates of different subspace clustering algorithms. As the table reveals, deep subspace clustering methods, including DSC, ADSC,

Table 5.3: Clustering error (%) of different methods on Extended Yale B, ORL, COIL20, and COIL100 datasets. Top performers are bolded.

dataset	LRR [34]	LRSC [155]	SSC [33]	AE+SSC [4]	KSSC [108]	SSC-OMP [117]	EDSC [119]	AE+EDSC [119]	DSC [4]	DASC [65]	S^2ConvSCN [64]	MLRDSC [66]	MLRDSC-DA Ours
E.Yale B	34.87	29.89	27.51	25.33	27.75	24.71	11.64	12.66	2.67	1.44	1.52	1.36	**0.82**
ORL	33.50	32.50	29.50	26.75	34.25	37.05	27.25	26.25	14.00	11.75	10.50	11.25	**10.25**
COIL20	30.21	31.25	14.83	22.08	24.65	29.86	14.86	14.79	5.42	3.61	2.14	2.08	**1.79**
COIL100	53.18	50.67	44.90	43.93	47.18	67.29	38.13	38.88	30.96	–	26.67	23.28	**20.67**

Table 5.4: Clustering error (%) on Extended Yale B with different augmentation policies applied to the inputs of MLRDSC-DA.

Augmentation Policies:	Random LR Flips	Cut-out	Common aug. policies	AutoAug for ImageNet	AutoAug for SVHN	Policies found from Algorithm 1 (ours)
Extended YaleB	1.32	2.88	2.96	5.96	11.31	**0.82**

S^2ConvSCN, and ML-RDSC, in general, outperform the conventional subspace clustering approaches. This observation suggests that deep networks can better model the non-linear relationships between the samples. However, among them, our model outperforms all the benchmarks. Note that our model and MLRDSC share similar architectures and have the same number of parameters. The only difference is that our method takes advantage of training on the augmented set of data. This observation clearly shows the benefits of incorporating data augmentation in the task of deep subspace clustering.

5..4 Comparison with Common Augmentation Policies and Transferred Augmentation Policies

Existing automated learning algorithms for finding proper augmentations or even manual searches do not apply to the subspace clustering task. The current algorithms are mostly designed for supervised tasks and require the ground-truth targets to compare the performances, whereas, in the subspace clustering task, the ground-truth labels are not available. However, one may apply the supervised augmentation searches to a source dataset with available labels and use the found augmentation policies on a target dataset for the task of subspace clustering.

To compare such an approach with the described method in Section 4., we adopt the augmentation policies that AutoAug [140] finds on the classification task for SVHN [12] and ImageNet [156] datasets, and directly apply the found policies to the input of our MLRDSC-DA.

We furthermore compare the performances to the results of applying the following augmentation policies to the input: random left-right flips (*Flip-LR*), *Cut-out* [157, 158] and common augmentations picked by practitioners (*Common aug. policies*). For "Common aug. policies", we use the combination of most common augmentations, including zero paddings, cropping, random-flips, and cutout.

Note that all the experiments in this section share the same architecture and training procedure as MLRDSC-DA. They are only different in the augmentation policies that are applied to their input.

As can be seen in Table 5.4, the augmentation policies that are found with [140] on SVHN and ImageNet, perform poorly. This is because they are deemed good policies for the classification task on those datasets and may not work as efficiently on the subspace clustering task. The reason that Random Flips provides a relatively good performance is that the objects in the dataset are symmetric. The augmentations that are found with our suggested approach provide the best results.

6. Conclusion

We introduced a framework to incorporate data augmentation techniques in Deep Subspace Clustering algorithms. The underlying assumption in subspace clustering tasks is that data points with the same label lie into the same subspace. Based on this assumption, we argued that slight transformations of a data point should not alter the subspace into which the data point lies. To address this property, we proposed the subspace consistency loss to keep the data points within consistent subspaces when slight random transformations are applied to the input data. Employing the mean Silhouette coefficient metric, we furthermore, provided a simple yet effective unsupervised algorithm to find the best augmentation policies for each target dataset. Our experiments showed that applying good data augmentations improves the performance oft subspace clustering algorithms.

7. Broader Impact

Since our method improves subspace clustering, it advances learning from unannotated data. Improving the learning process and providing more accurate similarity matrices for unannotated data can positively impact accountability, transparency and explainability of AI methods. However, if not controlled, providing the opportunity to learn from big unannotated datasets could increase the concerns about violating the privacy of individuals.

Part II

Classification Tasks

Chapter 6

Deep Sparse Representation-based Classification

1. Introduction

Sparse coding has become widely recognized as a powerful tool in signal processing and machine learning with various applications in computer vision and pattern recognition [159, 160, 36]. Sparse representation-based classification (SRC) as an application of sparse coding was first proposed in [159], and was shown to provide robust performance on various face recognition datasets. Since then, SRC has been used in numerous applications [161, 162, 163, 164]. In SRC, an unlabeled sample is represented as a sparse linear combination of the labeled training samples. This representation is obtained by solving a sparsity-promoting optimization problem. Once the representation is found, the label is assigned to the test sample based on the minimum reconstruction error rule [159].

The SRC method is based on finding a linear representation of the data. However, linear representations are almost always inadequate for representing non-linear structures of the data which arise in many practical applications. To deal with this issue, some works have exploited the kernel trick to develop non-linear extensions of the SRC-based methods [165, 166, 167, 168, 169, 105, 170, 107, 171, 172, 39]. Kernel SRC methods require the use of a predetermined kernel function such as polynomial or Gaussian. Selection of the kernel function and its parameters is an important issue in training when kernel SRC methods are used for classification.

In this chapter, we propose a deep neural network-based framework that finds an explicit nonlinear mapping of data, while simultaneously obtaining sparse codes that can be used for classification. Learning nonlinear mappings with neural networks has been shown to produce remarkable improvements in subspace clustering tasks [4, 51]. We introduce a transductive model, which accepts a set of training and test samples, learns a mapping that is suitable for

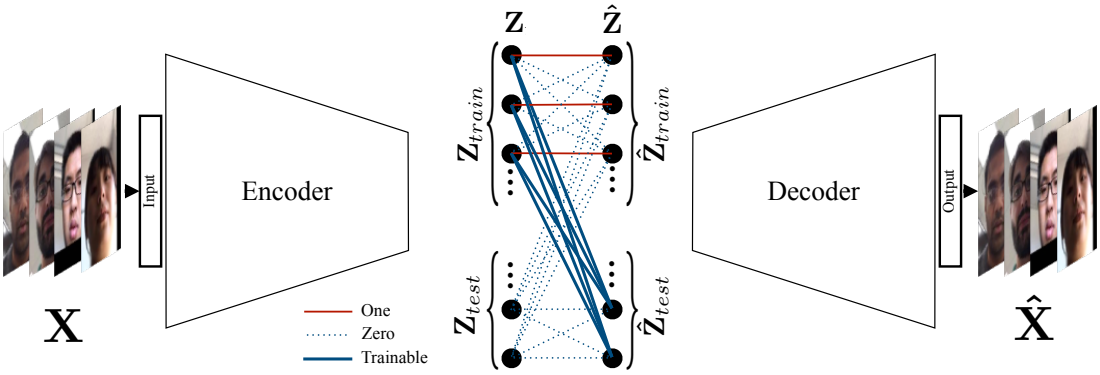


Figure 6.1: An overview of the proposed deep SRC network. The trainable parameters of sparse coding layer are depicted with solid blue lines. Note that $\mathbf{Z}_{train} = \hat{\mathbf{Z}}_{train}$, and $\mathbf{Z}_{test} \approx \hat{\mathbf{Z}}_{test} = \mathbf{Z}_{train}\mathbf{A}$.

sparse representation, and recovers the corresponding sparse codes. Our model consists of an encoder that is responsible for learning the mapping, a sparse coding layer which mimics the task of constructing the mapped test samples by a combination of the mapped training samples, and a decoder that is used for training the networks.

1..1 Sparse representation-based classification

In SRC, given a set of labeled training samples, the goal is to classify an unseen set of test samples. Suppose that we collect all the vectorized training samples with the label i in the matrix $\mathbf{X}^i_{train} \in \mathbb{R}^{d_0 \times n_i}$, where d_0 is the dimension of each sample and n_i is the number of samples in class i, then the training matrix can be constructed as

$$\mathbf{X}_{train} = [\mathbf{X}^1_{train}, \mathbf{X}^2_{train}, \cdots, \mathbf{X}^K_{train}] \in \mathbb{R}^{d_0 \times n} \tag{6.1}$$

where $n_1 + n_2 + \cdots + n_K = n$ and we have a total of K classes.

In SRC, it is assumed that an observed sample $\mathbf{x}_{test} \in \mathbb{R}^{d_0}$ can be well approximated by a linear combination of the samples in $\mathbf{X}^i_{train}$ if $\mathbf{x}_{test}$ is from class i. Thus, it is possible to predict the class of a given unlabeled data by finding a set of samples in the training set that can better approximate $\mathbf{x}_{test}$. Mathematically, these samples can be found by solving the following optimization problem

$$\min_{\boldsymbol{\alpha}} \|\boldsymbol{\alpha}\|_0 \quad \text{s.t.} \quad \mathbf{x}_{test} = \mathbf{X}_{train}\boldsymbol{\alpha}, \tag{6.2}$$

where $\|\boldsymbol{\alpha}\|_0$ counts the number of non-zero elements in $\boldsymbol{\alpha}$. The minimization problem (7.1) finds a sparse solution for the linear system. However, since the optimization problem (7.1) is

an NP-hard problem, in practice, a sparsity constraint is enforced by the ℓ_1-norm of $\boldsymbol{\alpha}$ which is a convex relaxation of the above problem [70, 71]. Thus, in practice the following minimization problem is solved to obtain the sparse codes

$$\min_{\boldsymbol{\alpha}} \|\mathbf{x}_{test} - \mathbf{X}_{train}\boldsymbol{\alpha}\|_2^2 + \lambda_0\|\boldsymbol{\alpha}\|_1, \tag{6.3}$$

where λ_0 is a positive regularization parameter. Once $\boldsymbol{\alpha}$ is found, one can estimate the class label of $\mathbf{x}_{test}$ as follows

$$\text{class}(\mathbf{x}_{test}) = \arg\min_k \|\mathbf{x}_{test} - \mathbf{X}_{train}\delta_k(\boldsymbol{\alpha})\|_2^2, \tag{6.4}$$

where $\delta_k(\cdot)$ is the characteristic function that selects the coefficients associated with the class i.

2. Deep sparse representation-based classification network

We develop a transductive classification model based on sparse representations. In a transductive model, as opposed to inductive models, both training and test sets are observed, and the learning process pursues reasoning from the specific training samples to a specific set of test cases [173]. We build our method based on convolutional autoencoders. In particular, our network contains an encoder, a sparse coding layer, and a decoder. The encoder receives both the training and test sets as raw data inputs and extracts abstract features from them. The sparse coding layer recovers the test cases by a sparse linear combination of the training samples, and concatenates them along with the training features which are then fed to the decoder. The decoder maps both the training embeddings and the recovered test embeddings back to the original representation of the data. Figure 6.1 gives an overview of the proposed deep SRC (DSRC) framework.

Sparse representation: Let $\mathbf{X}_{train} \in \mathbb{R}^{d_0 \times n}$ and, $\mathbf{X}_{test} \in \mathbb{R}^{d_0 \times m}$ be the given vectorized training and testing data, respectively. We feed $\mathbf{X} = [\mathbf{X}_{train}, \mathbf{X}_{test}]$ to the encoder, where it develops the corresponding embedding features $\mathbf{Z} = [\mathbf{Z}_{train}, \mathbf{Z}_{test}] \in \mathbb{R}^{d_z \times (m+n)}$. The minimization problem (7.2) for a single test observation can be re-written for a matrix of testing embedding features as

$$\min_{\mathbf{A}} \|\mathbf{Z}_{test} - \mathbf{Z}_{train}\mathbf{A}\|_F^2 + \lambda_0\|\mathbf{A}\|_1, \tag{6.5}$$

where $\mathbf{A} \in R^{n \times m}$ is the coefficient matrix that contains the sparse codes in its columns, and λ_0 is a positive regularization parameter. Note that the first penalty term in equation (7.7) is equivalent to the penalty term used for a fully-connected neural network layer with the input of $\mathbf{Z}_{train}$, the output of $\mathbf{Z}_{test}$ and trainable parameters of $\mathbf{A}$. As a result, considering the sparsity constraint, one can model the optimization problem (7.7) in a neural network framework with a fully-connected layer with sparse parameters which have no non-linearity activation or bias nodes. We use such a model inside our sparse coding layer to find the sparse codes for the observed test set.

The sparse coding layer is located between the encoder and decoder networks. Its task for $\mathbf{Z}_{train}$ is to pass them to the decoder, and for the test features $\mathbf{Z}_{test}$ it will pass their reconstructions that are found from (7.7), as $\mathbf{Z}_{train}\mathbf{A}$, to the decoder. Thus, assuming that $\hat{\mathbf{Z}}_{train}$ and $\hat{\mathbf{Z}}_{test}$ are the outputs of the sparse coding layer for training and testing features, we have

$$\hat{\mathbf{Z}}_{train} = \mathbf{Z}_{train}\mathbf{I}_n, \quad \hat{\mathbf{Z}}_{test} = \mathbf{Z}_{train}\mathbf{A}, \tag{6.6}$$

where $\mathbf{I}_n \in \mathbb{R}^{n \times n}$ is the identity matrix. Therefore, if the decoder's input is $\hat{\mathbf{Z}} = [\hat{\mathbf{Z}}_{train}, \hat{\mathbf{Z}}_{test}]$, from (7.8) we can calculate $\hat{\mathbf{Z}}$ as $\hat{\mathbf{Z}} = \mathbf{Z}\mathbf{\Theta}_{sc}$, where

$$\mathbf{\Theta}_{sc} = \begin{bmatrix} \mathbf{I}_n & \mathbf{A} \\ \mathbf{0}_{n \times m} & \mathbf{0}_m \end{bmatrix}. \tag{6.7}$$

In equation (7.9), $\mathbf{0}_{n \times m} \in \mathbb{R}^{n \times m}$ and $\mathbf{0}_m \in \mathbb{R}^{m \times m}$ are zero matrices. One can write an end-to-end training objective that includes sparse coding and training of the encoder-decoder as follows

$$\min_{\mathbf{\Theta}} \|\mathbf{Z} - \mathbf{Z}\mathbf{\Theta}_{sc}\|_F^2 + \lambda_0 \|\mathbf{\Theta}_{sc}\|_1 + \lambda_1 \|\mathbf{X} - \hat{\mathbf{X}}\|_F^2, \tag{6.8}$$

where $\mathbf{\Theta}$ is the union of all the trainable parameters including encoder and decoder's parameters and $\mathbf{A}$. Here, $\hat{\mathbf{X}} = [\hat{\mathbf{X}}_{train}, \hat{\mathbf{X}}_{test}]$ is the output of the decoder (i.e. reconstructions), and λ_0 and λ_1 are positive regularization parameters. Note that the optimization problem (7.10) simultaneously finds sparse codes $\mathbf{A}$ and a set of desirable embedding features $\mathbf{Z}$ that are especially suitable for providing efficient sparse codes.

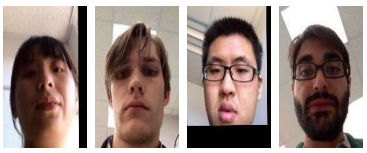
Classification: Once the sparse coefficient matrix $\mathbf{A}$ is found, it can be used for associating the class labels to the test samples. For each test sample $\mathbf{x}_{test}^i$ in $\mathbf{X}_{test}$, its embedding features

Algorithm 3 Deep sparse representation-based classification

procedure DSRC($\mathbf{X}_{train}, \mathbf{X}_{test}, \lambda_0, \lambda_1$).
    Construct $\mathbf{X} = [\mathbf{X}_{train}, \mathbf{X}_{test}]$.
    Find $\mathbf{A}$ via $\mathbf{\Theta}$ by solving the optimization problem (7.10).
    Classify the test samples using (6.9) .
end procedure

(a) USPS [6]

(b) SVHN [12]



(c) UMDAA-01 [13]

Figure 6.2: Sample images from (a) USPS [6], (b) SVHN [12], and (c) UMDAA-01 [13].

$\mathbf{z}^i_{test}$, and the corresponding sparse code column $\boldsymbol{\alpha}^i$ in $\mathbf{A}$ are used to estimate the class labels as follows

$$\text{class}(\mathbf{x}^i_{test}) = \arg\min_k \|\mathbf{z}^i_{test} - \mathbf{Z}_{train}\delta_k(\boldsymbol{\alpha}^i)\|_2^2. \tag{6.9}$$

The proposed DSRC method is summarized in Algorithm 3.

3. Experimental results

In this section, we evaluate our method against state-of-the-art SRC methods. The USPS handwritten digits dataset [6], the street view house numbers (SVHN) dataset [12], and the UMDAA-01 face recognition dataset [13] are used in our experiments. Figure 8.5 (a), (b), and (c) show sample images from these datasets. Since the number of parameters in the sparse coding layer scales with the multiplication of training and testing sizes, we randomly select a smaller subset of the used datasets and perform all the experiments on the selected subset. In all the experiments, the input images are resized to 32×32.

We compare our method with the standard SRC method [159], Kernel SRC (KSRC) [169], SRC on features extracted from an autoencoder with similar architecture to our network (AE-SRC), and SRC on features extracted from the state-of-the-art pre-trained networks. In our experiment with the pre-trained networks, the networks are pre-trained on the Imagenet dataset [174]. For this purpose, we use the following four popular network architectures: VGG-19[175],

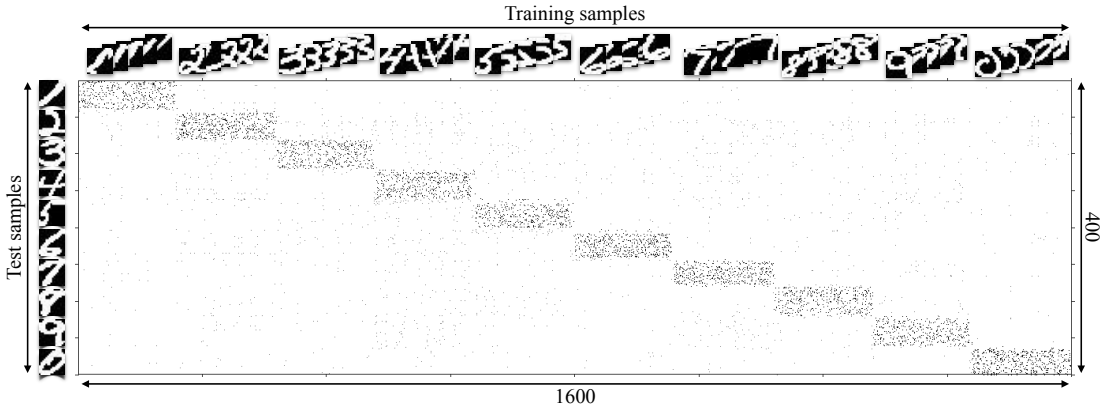


Figure 6.3: Visualization of the sparse coding matrix ($\mathbf{A}$) in the experiment with the USPS dataset. Note that for better visualization the absolute value of the transposed $\mathbf{A}$ (i.e. $|\mathbf{A}^T|$) is shown.

Inception-V3 [176], Resnet-50 [177] and Densenet-169 [178]. We feed these networks with our datasets, extract the features corresponding to the last layer before classification, and pass them to the classical SRC algorithm. Note these networks accept 224×224 inputs. Thus, as a preprocessing step, we resample the input images to 224×224 images before feeding them to the pre-trained networks.

We compare different methods in terms of their five-fold averaged classification accuracy. In all the experiments, unless otherwise stated, we randomly split the datasets into sets of training and testing samples, where 20% of the samples are used for testing, and 80% of the samples are used as the training set.

Network structure: The encoder network of our model consists of stacked four convolutional layers, and the decoder has three fractionally-strided convolution layers (also known as deconvolution layers). Each plugged in convolution or fractionally-strided convolution is coupled with a ReLu nonlinearity as well, but does not have a batch-norm layer. Table 6.1 gives the details of the network, including the kernel sizes and the number of filters.

Training details: We implemented our method with Tensorflow-1.4 [128]. We use the adaptive momentum-based gradient descent method (ADAM) [129] to minimize the loss function, and apply a learning rate of 10^{-3}. Before we start training on our objective function, in each experiment, we pre-train our encoder and decoder on the dataset without the sparse coding layer. In particular, we pre-train the encoder-decoder for $20k$ epochs with the objective of

Table 6.1: Details of our networks. Note that the number of parameters in the sparse coding layer rely on the size of dataset including the *n* training and *m* test samples.

	Layer	Input	Output	Kernel	(stride, pad)
Encoder	Conv 1	$\mathbf{X}$	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	$\mathbf{Z}$	$1 \times 3 \times 3 \times 30$	(1,0)
Sparse coding layer	$\mathbf{\Theta}_{sc}$	$\mathbf{Z}$	$\hat{\mathbf{Z}}$	$m \times n$ Parameters	-
Decoder	deconv 1	$\hat{\mathbf{Z}}$	deconv 1	$1 \times 3 \times 3 \times 30$	(1,0)
	deconv 2	deconv 1	deconv 2	$1 \times 3 \times 3 \times 20$	(2,1)
	deconv 3	deconv 2	$\hat{\mathbf{X}}$	$1 \times 5 \times 5 \times 10$	(2,1)

Dataset	SRC	KSRC	AE-SRC	VGG19-SRC	InceptionV3-SRC	Resnet50-SRC	Denesnet169-SRC	DSRC (ours)
USPS	87.78	91.34	88.65	91.27	93.51	95.75	95.26	**96.25**
SVHN	15.71	27.42	18.69	52.86	41.14	47.88	37.65	**67.75**
UMDAA-01	79.00	81.37	86.70	82.68	86.15	91.84	86.35	**93.39**

Table 6.2: Sparse representation-based classification accuracy of different methods.

$\min_{\hat{\mathbf{\Theta}}} \|\mathbf{X} - \hat{\mathbf{X}}\|_F^2$, where $\hat{\mathbf{\Theta}}$ indicates the union of parameters in the encoder and decoder networks. We use a batch size of 100 for this stage. However, in the actual stage of training, we feed all the samples including the training and testing samples as a single large batch. We set the regularization parameters as $\lambda_0 = 1$ and $\lambda_1 = 8$ in all the experiments.

3..1 USPS digits

The first set of experiments is conducted on the USPS handwritten digits dataset [6]. This dataset contains 7291 training and 2007 test grayscale images of ten digits (0-9). Figure 8.5 (a), shows example images from this dataset. We perform the experiments on a subset with a total size of 2000 samples. In particular, we randomly select 160 and 40 samples per digit from the training and testing sets, respectively. The first row of Table 6.2 shows the performance of various SRC methods. As can be observed from this table, the proposed method performs significantly better than the other methods including the classical and deep learning-based methods.

Figure 6.3 shows the coefficient matrix $\mathbf{A}$, extracted from $\mathbf{\Theta}_{\boldsymbol{sc}}$, the matrix of the network trained for this experiment. For better visualization, we show the absolute value of the transposed $\mathbf{A}$ (i.e. $|\mathbf{A}^T|$). Thus, each row of the matrix in Figure 6.3 corresponds to the sparse codes for one of the test samples. Similarly, columns in this figure are coefficients related to

	DSRC	DSC-SRC	$\text{DSRC}_{0.5}$	$\text{DSRC}_{1.5}$	DSRC_{2}
USPS	**96.25**	78.25	N/C	95.75	**96.25**

Table 6.3: The classification accuracy corresponding to the ablation study. N/C refers to the cases where the learning process did not converge.

the training samples. This matrix is sparse and shows a block diagram pattern, where most of the non-zero coefficients for each test sample are those that correspond to the training samples with the same class as the observed test sample.

Analysis of the network: To understand the effects of some of our model choices, we compare the performance of our DSRC method with variations of it by changing the regularization norm on $\boldsymbol{\Theta_{sc}}$ in the loss function (7.10). We replace the term $\|\boldsymbol{\Theta}_{sc}\|_1$ in (7.10) by $\|\boldsymbol{\Theta}_{sc}\|_p$, where $p = 0.5, 1.5$ and 2, and report their performances by $DSRC_{0.5}$, $DSRC_{1.5}$ and $DSRC_2$, respectively.

In addition, if we do not follow the specific structure described in equation (7.9), and instead have a fully connected layer with $(m+n)^2$ parameters which receives $\mathbf{Z}$ and reconstructs $\hat{\mathbf{Z}}$, the architecture of the network will be similar to the deep subspace clustering networks (DSC) proposed in [4] for the task of subspace clustering. As an ablation study, we use this method to extract sparse codes and then apply the same classification rule as in (6.9) to estimate class labels for the test set. We call this method *DSC-SRC*.

Table 8.7 reveals that while the regularization norm on the coefficient matrix is selected between ℓ_1 and ℓ_2, it does not have much effect on the performance of the classification task. However, in our experiments, we observed that for norms smaller than 1, the problem is not stable and often does not converges. In addition, *DSC-SRC* cannot provide a desirable performance. Note that the fully-connected layer in this method (counterpart to our sparse coding layer) does not limit the testing features to be reconstructed with only the training features. As a result, it is possible that testing features shape an isolated group that does not have a strong connection to the training features. This makes it more difficult to estimate a label for the test samples.

3..2 Street view house numbers

The SVHN dataset [12] contains 630,420 color images of real-world house numbers collected from Google Street View images. This dataset has three splits as the training set with 73,257 images, the testing set with 26,032 images and an extra set containing 531,131 additional samples. In this experiment, similar to our experiments on MNIST, we randomly select 160 images per digit from the training split and 40 per digit from the test split. This dataset is much more challenging than MNIST. This is in part due to the large variations of data. Furthermore, many samples in this dataset contain multiple digits in an image. The task is to classify the center digit.

The second row in Table 6.2 compares the performance of different SRC methods. This table demonstrates the advantage of our method. While the classification task is much more challenging on SVHN than MNIST, the gap between the performance of our method and the second best performance is even more. The next best performing method is *VGG19-SRC* which performs 14.86% behind the accuracy of our method.

3..3 UMD mobile faces

The UMD mobile face dataset (UMDAA-01) [13] contains 750 front-facing camera videos of 50 users captured while using a smartphone. This dataset has been collected over three different sessions. This dataset was originally collected for the active authentication task, but since its frames include challenging facial image instances with various illumination and pose conditions it has also been used for other tasks [179, 137]. In this experiment, we randomly select 50 facial images per subjects from the data in Session 1. Figure 8.5 shows some sample images from this dataset.

The performance of various SRC methods on the UMDAA-01 dataset are tabulated in the third row of Table 6.2. As can be seen, our proposed DSRC method similar to the experiments with SVHN provides remarkable improvements as compared to the other SRC methods. This clearly shows that more challenging datasets are better represented by our method. This is because our method not only efficiently finds the sparse codes, but also it seeks for a representation of data (the output of the encoder) that is especially suitable for sparse representation.

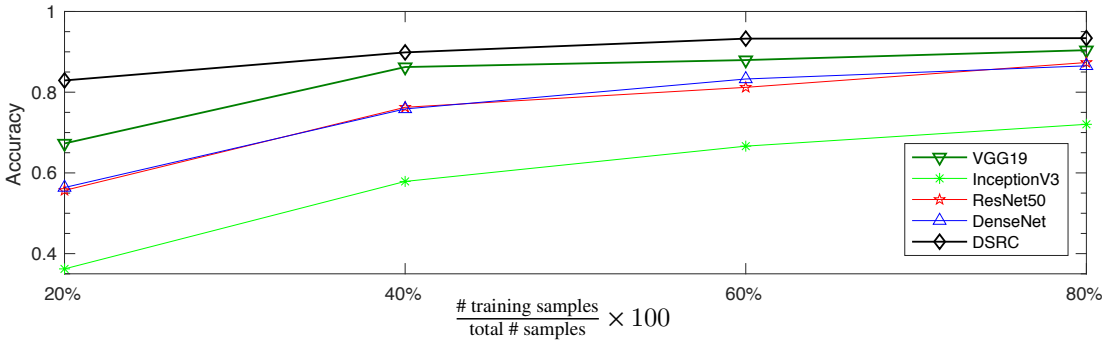


Figure 6.4: Effect of the number of training samples on the performance of different classification networks. The figure shows five-fold averaged classification accuracies of the methods trained on varying number of training samples in the UMDAA-01 dataset.

Comparison to state-of-the-art classification networks: While deep neural networks perform very well when they are trained on large datasets, in the case of limited number of labeled training samples, they often tend to overfit to the training samples. The objective of this experiment is to analyze the performance of our approach in such circumstances. We compare our method to the following classification networks: VGG-19[175], Inception-V3 [176], Resnet-50 [177] and Densenet-169 [178]. We first pre-train the networks on the Imagenet dataset [174], and then fine-tune them on the available training samples in UMDAA-01.

Figure 6.4 shows the performance of the classification networks on four different versions of UMDAA-01 dataset with varying number of training samples. The four versions are created by randomly splitting the dataset into sets of training and testing samples that respectively contain 20%, 40%, 60% and 80% of the total number of samples as training samples and use the rest of samples as the testing set. As the figure suggests, accuracy improves by increasing the number of training samples in all the cases. However, the results show better performances for DSRC even when less training data is available.

4. Conclusion

We presented an autoencoder-based sparse coding network for SRC. In particular, we introduced a sparse coding layer that is located between the conventional encoder and decoder networks. This layer recovers sparse codes of embedding features that are received from the encoder. The spare codes are later used to estimate the class labels of testing samples. We discussed a framework that allows an end-to-end training. Various experiments on three different

image classification datasets showed that the proposed network leads to sparse representations that give better classification results than state-of-the-art SRC methods.

Chapter 7

Deep Multimodal Sparse Representation-based Classification

1. Introduction

Sparse representation is an established technique in signal and image processing with various applications [159, 160, 36, 164]. Among the many applications, Sparse Representation Classification (SRC) methods exploit the discriminative nature of sparse codes and provide robust classification models less sensitive to non-constant variability, outliers, and small data sets [159, 162, 180, 181]. The SRC method uses a sparsity-promoting optimization problem to represent an unlabeled test sample as a sparse linear combination of labeled training samples. This representation is then used in assigning a label to the test sample based on the minimum reconstruction error rule [159]. Various SRC-based methods have been proposed in the literature. These methods include linear models in various applications [162, 180], kernel trick-based nonlinear models [169, 105, 170], and a recent deep neural network-based SRC method (DSRC) that finds an explicit nonlinear mapping for data, while simultaneously obtaining sparse codes that can be used for classification. Due to the efficiency of sparse coding-based algorithms, many works focus on designing specialized hardware to support sparse data [182].

Many real-world phenomena involve multiple modalities. Learning from multimodal sources offers the opportunity to gain an in-depth understanding of the phenomena by integrating the complementary information provided in different modalities [122, 120]. In multimodal learning, the model receives the data from multiple modalities and learns to fuse them. The information from different modalities can be fused at feature level (i.e., early fusion), decision level (i.e., late fusion), or intermediately [122, 120, 51].

In this chapter, we propose a multimodal deep SRC-based method. We enforce the different modalities to interact through the sparse coefficients of their latent space features. Our deep networks learn the latent space features of different modalities through an autoencoder

framework. Our framework encourages different modalities to learn latent features that are discriminative, suitable for sparse coding, and lie in mutual subspaces. The latent space features for the test samples are reconstructed by a linear combination of training samples with a sparse coefficient matrix that is shared among all the modalities.

Sharing the coefficient matrix among all the modalities pushes the test sample to interact with the training samples of all the modalities simultaneously. Therefore a more reliable coefficient matrix, which is calculated by the complementary information from different modalities, is constructed. Since the labels for the training samples are available, we use extra supervision based on labels to develop discriminative latent features. This extra supervision is employed by discriminator heads that are connected to latent space features of different modalities and are trained by a classification loss. At the test time, we combine the prediction of discriminator heads with the minimum reconstruction error rule, and introduce a new classification rule to assign labels to the test samples.

2. Related Work

Sparse Representation-based Classification:

In the SRC task, we are given a set of labeled training samples, and the goal is to classify an unseen set of testing samples. Suppose that we collect all the vectorized training samples in the matrix $\mathbf{X}_{\text{train}} \in \mathbb{R}^{d_0 \times n}$, where d_0 is the dimensional size of each sample, and n is the number of training samples.

SRC is based on the assumption that an observed sample $\mathbf{x}_{\text{test}} \in \mathbb{R}^{d_0}$ can be well approximated by a linear combination of samples in $\mathbf{X}_{\text{train}}$ that share the same class label as $\mathbf{x}_{\text{test}}$. Therefore, one can predict class label of a given unseen data such as $\mathbf{x}_{\text{test}}$ by finding the set of few samples in the training set that can better approximate $\mathbf{x}_{\text{test}}$. These samples can be picked out by solving the following optimization problem.

$$\min_{\boldsymbol{\alpha}} \|\boldsymbol{\alpha}\|_0 \;\; \text{s.t.} \;\; \mathbf{x}_{\text{test}} = \mathbf{X}_{\text{train}}\boldsymbol{\alpha}. \tag{7.1}$$

where $\|\boldsymbol{\alpha}\|_0$ counts the number of non-zero elements in $\boldsymbol{\alpha}$. In practice, the ℓ_0 norm is replaced by the ℓ_1 norm [70, 71]. Thus, in the case of noisy obseervations, the SRC solves the following

sparsity-promoting problem

$$\min_{\boldsymbol{\alpha}} \|\mathbf{x}_{\text{test}} - \mathbf{X}_{\text{train}}\boldsymbol{\alpha}\|_F^2 + \lambda_0\|\boldsymbol{\alpha}\|_1 \tag{7.2}$$

where λ_0 is a positive regularization parameter.

The solution, $\boldsymbol{\alpha}$, is used in the minimum reconstruction rule to estimate the class label of $\mathbf{x}_{\text{test}}$ as follows

$$\text{class}(\mathbf{x}_{\text{test}}) = \arg\min_k \|\mathbf{x}_{\text{test}} - \mathbf{X}_{\text{train}}\Gamma_k(\boldsymbol{\alpha})\|_F^2 \tag{7.3}$$

where $\Gamma_k(\cdot)$ is the matrix indicator function defined by setting all the rows but those corresponding to the ith class equal to zero.

Linear Multimodal Sparse Representation-based Classification: A number of multimodal extensions for the linear SRC problem have been proposed in the literature [162, 183, 184, 185, 186, 187]. Among those, the methods proposed in [162, 185, 187] are the closest to our model. They impose joint sparsities within and across different modalities. This way, the correlations and coupling the information among modalities are simultaneously taken into account.

Assume the given multimodal data is observed in M modalities, each with n training samples. For each modality $m = 1, 2, \cdots, M$, let's denote $\mathbf{X}_{\text{train}}^m \in \mathbb{R}^{d_m \times n}$ as the dictionary of training samples in m-th modality where d_m is the feature dimension of data in m-th modality. Similarly, the representation of a multimodal test sample in the m-th modality can be represented as $\mathbf{x}_{\text{test}}^m \in \mathbb{R}^{d_m}$.

The SRC model can be applied to the individual modalities. In other words, $\mathbf{x}_{\text{test}}^m$ can be reconstructed by a linear combination of a few atoms in the dictionary $\mathbf{X}_{\text{train}}^m$. Thus, we have

$$\mathbf{x}_{\text{test}}^m = \mathbf{X}_{\text{train}}^m\boldsymbol{\alpha}^m + \mathbf{N}^m, \tag{7.4}$$

where $\boldsymbol{\alpha}^m \in \mathbb{R}^n$ is a sparse coefficient vector, and $\mathbf{N}^m \in \mathbb{R}^{d_m \times n}$ is the noise matrix. The joint sparsity model argues that $\boldsymbol{\alpha}^m$ has the same sparsity pattern across the different modalities for $m = 1, 2, \cdots, M$. In other words, the matrix $\mathbf{A} = [\boldsymbol{\alpha}^1, \cdots, \boldsymbol{\alpha}^M]$ formed by concatenating the coefficient vectors of an observation across different modalities has the same non-zero rows in its different columns. The matrix $\mathbf{A}$ can be found by the following ℓ_1/ℓ_q-regularized least

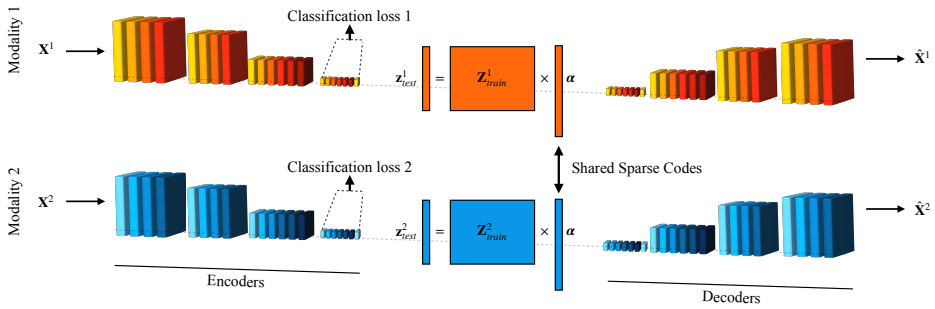


Figure 7.1: An overview of the proposed deep multimodal sparse representation-based classification network in a two-modality task. Features of different modalities are fed to their corresponding encoder, where a discriminative criterion is enforced to develop discriminative latent features that are especially suitable for jointly sparse representation. The latent features of different modalities are reconstructed by optimal joint sparse codes and are fed to decoders to reconstruct the raw modality features. The optimal joint sparse codes, along with the predictions of discriminator heads are exploited to predict the class labels of test samples.

square problem

$$\mathbf{A} = \arg\min_{\mathbf{A}} \frac{1}{2} \sum_{m=1}^{M} \|\mathbf{x}_{\text{test}}^m - \mathbf{X}_{\text{train}}^m \boldsymbol{\alpha}^m\|_F^2 + \lambda_0 \|\mathbf{A}\|_{1,q}. \tag{7.5}$$

where $q > 1$. Here, $\|\mathbf{A}\|_{1,q}$ is a norm defined as $\|\mathbf{A}\|_{1,q} = \sum_{k=1}^{n} \|\gamma^j\|_q$, where γ^j's are the rows in $\mathbf{A}$.

Once $\mathbf{A}$ is found, similar to problem (7.3), one can predict the class label of the test observation by solving the following problem

$$\min_k \|\mathbf{x}_{\text{test}}^m - \mathbf{X}_{\text{train}}^m \Gamma_k(\boldsymbol{\alpha}^m)\|_F^2 \tag{7.6}$$

3. Deep Multimodal Sparse Representation-based Classification Networks

Joint sparse representation-based classification methods [162, 185, 187] are able to extend the SRC model to a model that incorporates multiple modalities while keeping the benefits of sparse representation such as being less sensitive to outliers, and small data sets. However, they still rely on the assumption that the data points across different modalities show linear similarities within samples of the same class. This provides a strong motivation to incorporate deep neural networks to capture complex underlying structures of data across different modalities. We bridge multimodal SRC models and deep neural networks by proposing a transductive multimodal classification model based on deep sparse representation. A transductive model is a model in which both training and test sets are observed, and the learning process pursues

reasoning from the specific training samples to a specific set of test cases [173].

We use stacked multimodal autoencoders to exploit the nonlinear relations between the data points. In particular, we have a set of encoder and decoder per each available modality. The encoders and decoders are trained together to find latent space features that are discriminative, lie into a union of linear subspaces, and are constructed with the integrated information from all the available modalities.

To meet these properties, the autoencoder in each modality is trained according to both the training labels and the underlying structures of data in other modalities. The autoencoders of different modalities interact with each other by invoking the same linear relation between data points of different modalities in the training process. The same linear relation is imposed by sharing the same sparse codes in a sparsity-promoting reconstruction loss. As will be described in detail, the sparse codes can be modeled with a fully-connected layer in the deep neural networks framework. Figure 7.1 shows an overview of our framework.

Thus, the training objective of our model can be divided into reconstruction criteria and discriminative criterion.

Reconstruction Criteria: Reconstruction criteria itself consists of the reconstruction criterion in sparse coding and the reconstruction constraint for autoencoders.

Let $\{\mathbf{X}^m_{\text{train}}\}^M_{m=1}$ and $\{\mathbf{X}^m_{\text{test}}\}^M_{m=1}$ be the given set of training and test samples across the different modalities. We concatenate the training and test samples of each modality and construct the input matrices. The input matrix of m-th modality is denoted by $\mathbf{X}^m \in \mathbb{R}^{d_m \times n}$, and is constructed by the concatenation of $\mathbf{X}^m_{\text{train}} \in \mathbb{R}^{d_m \times n_{\text{train}}}$ and $\mathbf{X}^m_{\text{test}} \in \mathbb{R}^{d_m \times n_{\text{test}}}$, which are respectively the available training and testing samples in the m-th modality.

We feed $\mathbf{X}^m$ to its corresponding encoder and develop the embedding features $\mathbf{Z}^m$. The matrix $\mathbf{Z}^m$ consists of two types of embedding features. Those that are associated with the training samples and those that are corresponded to the testing samples. We respectively indicate to them with $\mathbf{Z}^m_{\text{train}}$ and $\mathbf{Z}^m_{\text{test}}$.

The SRC problem can be employed in the embedding feature space of each individual

modality. However, if we couple information among different modalities, richer representations can be learned. We propose to tie the embedding features of different modalities by enforcing them to share the same sparse code solutions in the SRC problem across the embedding space of all the different modalities. This way, the complementary information across different modalities are integrated without imposing an extra burden on the networks for explicitly learning to represent a joint representation.

Thus, we propose to find common sparse codes by solving the following optimization problem

$$\mathbf{A}_c = \arg\min_{\mathbf{A}_c} \frac{1}{2} \sum_{m=1}^{M} \|\mathbf{Z}^m_{\text{test}} - \mathbf{Z}^m_{\text{train}}\mathbf{A}_c\|_F^2 + \lambda_0 \|\mathbf{A}_c\|_1, \tag{7.7}$$

where $\mathbf{A}_c$ is the common sparse coding matrix. This matrix can be modeled by parameters of a set of M fully-connected layers with shared parameters. Note that the reconstruction term $\|\mathbf{Z}^m_{\text{test}} - \mathbf{Z}^m_{\text{train}}\mathbf{A}_c\|_F^2$ in the m-th modality is equivalent to the penalty term of a fully-connected layer with the input $\mathbf{Z}^m_{\text{train}}$, the output $\mathbf{Z}^m_{\text{test}}$ and the parameters $\mathbf{A}_c$. We use this in the implementation of our model and refer to the fully-connected layer as *joint sparse coding layer*.

The joint sparse coding layer is located between encoder and decoder of different modalities. This layer performs an identical task across all the modalities. It passes the training features to the corresponding decoder, and uses the parameters $\mathbf{A}_c$ to reconstruct the testing features, and passes the reconstructions to the decoders.

Assuming that $\hat{\mathbf{Z}}^m_{\text{train}}$ and $\hat{\mathbf{Z}}^m_{\text{test}}$ are respectively outputs of the common sparse coding layer for the training set and testing set in the m-th modality, we have

$$\hat{\mathbf{Z}}^m_{\text{train}} = \mathbf{I}_{n_{\text{train}}}\mathbf{Z}^m_{\text{train}}, \quad \hat{\mathbf{Z}}^m_{\text{test}} = \mathbf{Z}^m_{\text{train}}\mathbf{A}_c, \tag{7.8}$$

where $\mathbf{I}_{n_{\text{train}}} \in \mathbb{R}^{n_{\text{train}} \times n_{\text{train}}}$ is the identity matrix. Therefore, if for the mth decoder the input is $\hat{\mathbf{Z}}^m = [\hat{\mathbf{Z}}^m_{\text{train}}, \hat{\mathbf{Z}}^m_{\text{test}}]$, from (7.8) we can calculate $\hat{\mathbf{Z}}^m$ as $\hat{\mathbf{Z}}^m = \mathbf{Z}^m\mathbf{\Theta}_{sc}$, where

$$\mathbf{\Theta}_{sc} = \begin{bmatrix} \mathbf{I}_{n_{\text{train}}} & \mathbf{A}_c \\ \mathbf{0}_{n_{\text{train}} \times n_{\text{test}}} & \mathbf{0}_{\text{test}} \end{bmatrix}, \tag{7.9}$$

where $\mathbf{0}_{n_{\text{train}} \times n_{\text{test}}} \in \mathbb{R}^{n_{\text{train}} \times n_{\text{test}}}$ and $\mathbf{0}_{\text{test}} \in \mathbb{R}^{m \times m}$ are zero matrices.

Combining the criteria in sparse coding and training of the encoder-decoders, one can write

the reconstruction objective as

$$\mathcal{L}_{rec} = \sum_{m=1}^{M} \|\mathbf{Z}^m - \mathbf{Z}^m\mathbf{\Theta}_{sc}\|_F^2 + \lambda_0\|\mathbf{\Theta}_{sc}\|_1 + \sum_{m=1}^{M} \lambda_1\|\mathbf{X}^m - \hat{\mathbf{X}}^m\|_F^2 \tag{7.10}$$

Discriminative Criterion: We aim to train encoders by which the embeddings of different classes are best discriminated against. This property can be enforced on the encoders by incorporating labels for the training set. We plug discriminator heads to the output of encoders and train them to discriminate embedding features of different classes. Let $\mathbf{Y}$ represent the labels of the training samples; we define the discriminative criterion as follows

$$\mathcal{L}_{cls} = \sum_{m=1}^{M} \mathrm{CE}\left(D_m(\mathbf{Z}^m_{\text{train}}), \mathbf{Y}\right) \tag{7.11}$$

where D_m denotes the discriminator head that is dedicated to classifying the embedding features of m-th modality and $\mathrm{CE}(\cdot,\cdot)$ is the cross-entropy loss. We let the error of these discriminators be backpropagated to the encoders so that the encoders learn to produce separable embedding features.

$$\mathcal{L}_{cls} = \sum_{m=1}^{M} \|\mathbf{Z}^m - \mathbf{Z}^m\mathbf{\Theta}_{sc}\|_F^2 + \lambda_0\|\mathbf{\Theta}_{sc}\|_1 + \sum_{m=1}^{M} \lambda_1\|\mathbf{X}^m - \hat{\mathbf{X}}^m\|_F^2 \tag{7.12}$$

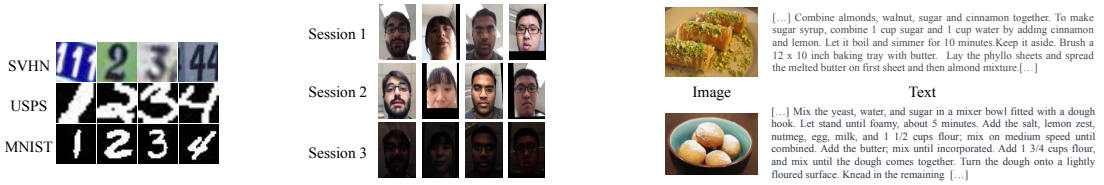
The discriminative criterion aims to train encoders that produce separable embedding features.

Full Objective: Combining the reconstruction and the discriminative criteria, our full objective function for training our networks is as follows:

$$\mathcal{L} = \mathcal{L}_{\text{rec}} + \beta\mathcal{L}_{\text{cls}}, \tag{7.13}$$

where $\beta > 0$ is a regularization parameter. Note that with our formulation, it is possible to train the networks in an end-to-end manner, and yet find the optimal sparse codes and encoder-decoder parameters, simultaneously.

Classification Rule: Once the networks are trained and the common sparse coding matrix $\mathbf{A}_c$ is found, we can use them for associating class labels to the test samples. Each test sample comes with m modalities as $\{\mathbf{x}^m_{\text{test}}\}_{m=1}^{M}$. They have the corresponding embedding features as



(a) (b) UMD-AA01 (c) UMPC-Food101

Figure 7.2: Samples from different modalities of datasets used in our experiments. (a) Digits from MNIST, SVHN and USPS. (b) Face images from different Sessions of UMDAA-01. (c) Food images and their recipe from UMPC-food101.

	SRC-C	J-SRC	score-fusion	feature-fusion	DMSRC (ours)	Unimodal SRC			Unimodal DSRC		
						M1	M2	M3	M1	M2	M3
Digits	91.87	92.34	18.25	18.13	96.25	11.98	88.13	92.25	62.75	94.25	95.37
Faces	83.45	84.76	14.37	15.17	94.13	78.32	77.21	75.83	91.21	90.56	89.12
Foods*	63.42	65.12	90.62	87.31	92.75	55.18	49.81	n/a	91.16	76.66	n/a

* All the methods for food-101 dataset use deep features extracted from DenseNet and BERT (for images and texts, respectively).

Table 7.1: Classification accuracy of different methods. M1 is SVHN in digits, Session1 in Faces and Images in Foods. M2 is USPS in digits, Session2 in Faces and texts in Foods. M3 is MNIST in digits, Session3 in Faces.

$\{\mathbf{z}_{\text{test}}^m\}_{m=1}^M$, and the corresponding sparse code column as $\boldsymbol{\alpha}$ in the common sparse code matrix $\mathbf{A}_c$. We can estimate the label of the sample by

$$\text{class}(\{\mathbf{x}_{\text{test}}^m\}_{m=1}^M) = \arg\min_k \sum_{m=1}^M \|\mathbf{z}_{\text{test}}^m - \mathbf{Z}_{\text{train}}^m \Gamma_k(\boldsymbol{\alpha})\|_F^2 \tag{7.14}$$

with $\Gamma_k(\cdot)$ similar to the equation (7.3).

In addition to the solution of (7.14), in our model, the discriminator heads in our framework can provide extra class label predictions for the test samples. Thus, we determine the final label estimates by ensembling the predictions of discriminator heads and the solution of (7.14). This is done by averaging the normalized scores.

4. Experimental results

We evaluate our method on three multimodal datasets for digit classification, face recognition, and food categorization. We evaluate our method against state-of-the-art unimodal SRC methods, multimodal SRC methods, and the commonplace fusion methods. In particular, we compare against SRC [159] and DSRC [**?**] as unimodal baselines. For multimodal SRCs, we compare against Joint Sparse representation (J-SRC) [162] as well as the classical SRC performed on the concatenation of individual modalities denoted as SRC-C. Finally, in the last

category of our baselines, we compare against the late feature fusion (feature-fusion), and score-fusion methods. Feature-fusion and score-fusion are of the most effective approaches in deep multimodal learning [122, 120].

We use the following datasets in our experiments:

Digits: We combine SVHN [12], USPS [6] and MNIST [5] digits datasets to assemble a multiview digit dataset. Here, we view images from the individual datasets as different views of the same digit. Since the number of parameters in the sparse coding layer of our model scales quadratically with the size of the data, we randomly select 200 samples per digit to keep the networks to a tractable size. In total, we have 2000 multiview digits.

Faces: We view different Sessions of the UMD mobile faces dataset (UMDAA-01) [13] as different modalities. We randomly select 50 facial images per subject from each Session.

UMPC Food-101 [188]: The dataset contains images of 101 different foods along with recipes found from the web for these datasets. We keep the first 10 classes and randomly select 200 samples per class in our experiments. For text normalization, we remove double spaces, lower case all characters, and remove any character other than the English alphabets.

Figure 8.5 (a), (b), and (c) show samples from the digits, UMDAA-01 and UMPC Food-101 datasets, respectively. We use 60% of the samples in each dataset as the training set, and the remaining 20% as the testing set.

Training details: We implemented our method with Tensorflow-1.4. We use the adaptive momentum-based gradient descent method (ADAM) [129] to minimize our loss functions, and apply a learning rate of 10^{-3}. Before we start training on our objective function, in each experiment, we pre-train our encoder and decoder on the dataset without the sparse coding layer. We set the regularization parameters as $\lambda_0 = 1$, $\lambda_1 = 8$ and $\beta = 1000$ in all the experiments.

4..1 Digits and Faces

For Digits and Faces datasets, we adopt the same architecture as described in [**?**]. That is using stacked autoencoders of four convolutional layers for the encoder and three deconvolution layers for the decoder per each modality. The first two rows in Table 7.1 compare the performance of our method against unimodal and multimodal classifiers on digits and faces datasets.

In the first row of Table 7.1, M1, M2, and M3 refer to SVHN, MNIST, and USPS datasets, respectively. In the second row, M1, M2 and M3 respectively refer to Session 1, Session 2 and Session 3 of UMDAA-01.

We observe multimodal SRC-based methods outperform the unimodal methods. This clarifies the benefits of integrating multiple modalities. However, score-fusion and feature fusion perform poorly here since the networks are shallow and are trained from scratch. Our DMSRC provides the best performance in both the datasets by using both the benefits of deep multimodal learning and the robustness of SRC-based methods.

4..2 Deep Networks with State-of-the-art Architectures

In this experiment, we evaluate our method against state-of-the-art deep neural networks. We adopt DenseNet [178] and BERT [189] networks that are of the most efficient deep architectures for processing images and texts, respectively. We use Wikipedia pre-trained BERT and pre-train DenseNet on Imagenet.

For both the networks, we add a fully-connected layer with 100 hidden nodes before the final classifier layer to provide a low-dimensional in-depth feature space in which the experiments of SRC-based methods are conducted. This layer is fine-tuned by the training samples of our UMPC-Food101 subset. Score-fusion and feature-fusion methods also use the same architecture.

For unimodal DSRC and our method, we use two fully-connected layers with 40 hidden state nodes as encoder and decoder.

In Table 7.1, we refer to the image modality as M1 and denote the text modality as M2. The results of Table 7.1 are interesting to compare with unimodal accuracies of 90.12% for DenseNet in the image modality, and 72.33% for BERT in the text modality. As Table 7.1 reveals, score-fusion and feature-fusion methods perform quite well, and on contrast, linear SRC-based method does not show a strong performance. It shows that although the in-depth features provide discriminative features, the learned features are not suitable for a linear sparse representation. DSRC and our MDSRC, on the other hand, can successfully map the features to a latent space in which the benefits of SRC-based methods can be exploited. Our MDSRC, outperforms all the baselines.

5. Conclusion

We presented a deep sparse representation-based fusion method for classifying multimodal signals. We use autoencoders to develop features that may be different across different modalities but share the same sparse codes in sparse representation classification problems that are applied to separate modalities. The training objective for encoders consists of reconstruction criteria and discriminative criterion. We proposed a classification rule that uses sparse codes as well as the prediction of classification heads in different modalities to determine an accurate estimate for classifying the test samples.

Chapter 8

Improving the Performance of Unimodal Dynamic Hand-Gesture Recognition with Multimodal Training

1. Introduction

Recent advances in computer vision and pattern recognition have made hand gesture recognition an accessible and important interaction tool for different types of applications including human-computer interaction [190], sign language recognition [191], and gaming and virtual reality control [192]. In particular, recent developments in deep 3-D convolutional neural networks (3D-CNNs) with video sequences have significantly improved the performance of dynamic hand gesture recognition [193, 194, 3].

Most state-of-the-art hand gesture recognition methods exploit multiple sensors such as visible RGB cameras, depth camera or compute an extra modality like optical flow to improve their performances [195, 196, 197, 198]. Multimodal recognition systems offer significant improvements to the accuracy of hand gesture recognition [199]. A multimodal recognition system is trained with multiple streams of data and classifies the multimodal observations during testing [122] (Figure 8.1 (a)). On the other hand, a unimodal recognition system is trained and tested using only a single modality data (Figure 8.1 (b)). This chapter introduces a third type of framework which leverages the knowledge from multimodal data during training and improves the performance of a unimodal system during testing. Figure 8.1 (c) gives an overview of the proposed framework.

The proposed approach uses separate 3D-CNNs per each stream of modality for primarily training them to recognize the dynamic hand gestures based on their input modality streams. The streams of modalities that are available in dynamic hand gesture recognition systems are often spatially and temporally aligned. For instance, the RGB and depth maps captured with motion sensing devices and the optical flow calculated from the RGB streams are usually aligned.

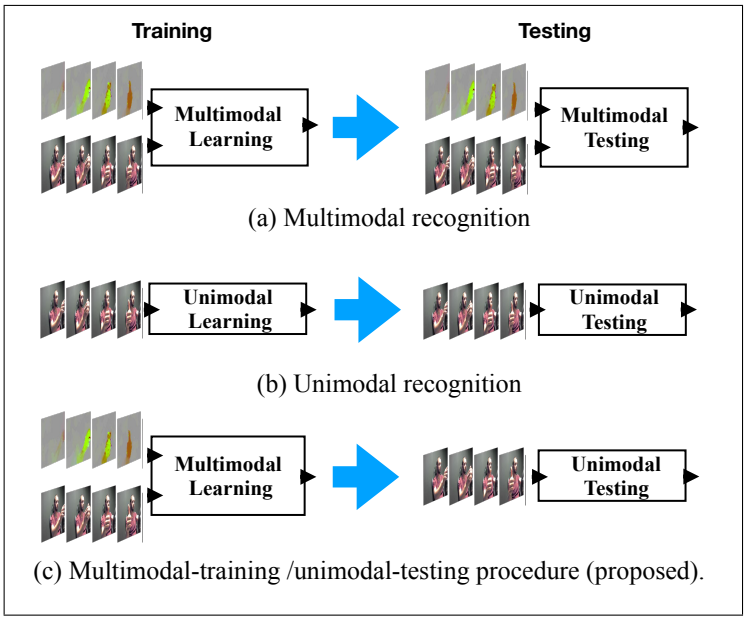


Figure 8.1: Training and testing schemes of different types of recognition systems. (a) The system is trained and tested with multiple modalities. (b) The system is trained and tested with a single modality. (c) The system leverages the benefits of multimodal training but can be ran as a unimodal system during testing.

Hence, we encourage the individual modality networks to derive a common understanding for the spatiotemporal contents of different modalities. We do this by sharing their knowledge throughout the learning process by minimizing the introduced *spatiotemporal semantic alignment (SSA)* loss.

We further improve the learning process by regularizing the SSA loss with an adaptive regularization parameter. We call this regularization parameter, the *focal regularization parameter*. This parameter prevents the transfer of negative knowledge. In other words, it makes sure that the knowledge is transferred from more accurate modality networks to less accurate networks and not the other way. Once the networks are trained, during inference, each network has learned to recognize the hand gestures from its dedicated modality, but also has gained the knowledge transferred from the other modalities that assists in providing the better performance.

In summary, this chapter makes the following contributions. First, we propose a new framework for single modality networks in dynamic hand gesture recognition task to learn from multiple modalities. This framework results in a *Multimodal Training / Unimodal Testing (MTUT)* scheme. Second, we introduce the *SSA* loss to share the knowledge of single modality networks. Third, we develop the *focal regularization parameter* for avoiding negative transfer. In our experiments, we show that learning with our method improves the test time performance of unimodal networks.

2. Related Work

Dynamic Hand Gesture Recognition: Dynamic hand-gesture recognition methods can be categorized on the basis of the video analysis approaches they use. Many hand-gesture methods have been developed based on extracting handcrafted features [200, 201, 202, 1]. These methods often derive properties such as appearance, motion cues or body-skeleton to perform gesture classification. Recent advances in action recognition methods and the introduction of various large video datasets have made it possible to efficiently classify unprocessed streams of visual data with spatiotemporal deep neural network architectures [15, 203, 204].

Various 3D-CNN-based hand gesture recognition methods have been introduced in the literature. A 3D-CNN-based method was introduced in [194] that integrates normalized depth and image gradient values to recognize dynamic hand gestures. In [199], a 3D-CNN was proposed that fuses streams of data from multiple sensors including short-range radar, color and depth sensors for recognition. A real-time method is proposed in [3] that simultaneously detects and classifies gestures in videos. Camgoz et al. [205] suggested a user-independent system based on the spatiotemporal encoding of 3D-CNNs. Miao et al. proposed ResC3D [198], a 3D-CNN architecture that combines multimodal data and exploits an attention model. Furthermore, some CNN-based models also use recurrent architectures to capture the temporal information [196, 2, 206, 207].

The main focus of this chapter is to improve the performance of hand gesture recognition methods that are built upon 3D-CNNs. As will be described later, we assume that our networks have 4-D feature maps that contain positional, temporal and channel dimensions.

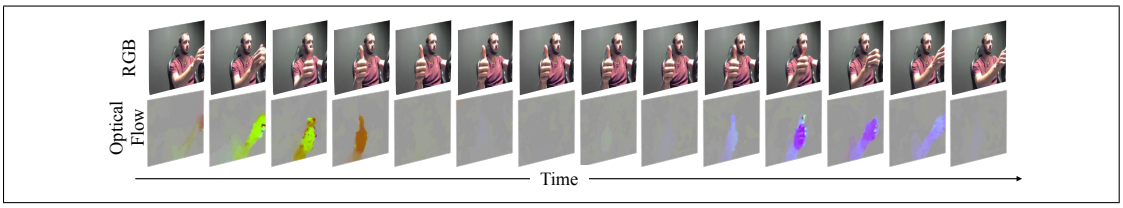


Figure 8.2: An example of the RGB and optical flow streams from the NVGesture Dataset [3]. As can be seen, while for the stationary frames RGB provides better representation, optical flow provides better representation for the dynamic frames.

Transfer Learning: In transfer learning, first, an agent is independently trained on a source task, then another agent uses the knowledge of the source agent by repurposing the learned features or transferring them to improve its learning on a target task [208, 209]. This technique has been shown to be successful in many different types of applications [210, 211, 212, 213, 214, 215]. While our method is closely related to transfer learning, our learning agents (i.e. modality networks) are trained simultaneously, and the transfer occurs both ways among the networks. Thus, it is better categorized as a multi-task learning framework [216, 217], where each network has three tasks of providing the knowledge to the other networks, receiving the knowledge from them, and finally classifying based on their dedicated input streams.

Multimodal Fusion: In multimodal fusion, the model explicitly receives the data from multiple modalities and learns to fuse them [120, 49, 218]. The fusion can be achieved at feature level (i.e. early fusion), decision level (i.e. late fusion) or intermediately [122, 51]. Once the model is trained, during testing, it receives the data from multiple modalities for classification [122, 120]. While our method is related to multimodal fusion, it is not a fusion method. We do not explicitly fuse the representations from different modalities. Instead, we improve the representation learning of our individual modality networks by leveraging the knowledge from different modalities. During inference, we do not necessarily need multiple modalities but rather each individual modality network works independently to classify data.

3. Proposed Method

In our proposed model, per each modality, one 3D-CNN is trained. Assuming that the stream of data is available in M modalities, we have M classifier networks with similar architectures that

classify based on their corresponding input. During training, while each network is primarily trained with the data from its corresponding modality, we aim to improve the learning process by transferring the knowledge among the networks of different modalities. The transferred knowledge works as an extra supervision in addition to the class labels.

We share the knowledge of networks by aligning the semantics of the deep representations they provide for the inputs. We do this by selecting an in-depth layer in the networks and enforcing them to share a common correlation across the in-depth layers of all the modality networks. This is done by minimizing the distance between their correlation matrices in the training stage. In addition, we regularize this loss by an adaptive parameter which ensures that the loss is serving as a one-way gate that only transfers the knowledge from more accurate modality networks to those with less accuracy, and not the other way.

3..1 Spatiotemporal Semantic Alignment

In an ideal case, all the M classifier modality networks of our model should have the same understanding for an input video. Even though they are coming in different modalities, their inputs are representing the same phenomena. In addition, since we assume that different modalities of the input videos are aligned over the time and spatial positions, in an ideal case the networks are expected to have the same understanding and share semantics for spatial positions and frames of the input videos across the different modalities. However, in practice, some spatiotemporal features may be better captured in one modality as compared to some other modalities. For instance, in the stream of visible RGB and optical flow frames shown in Figure 8.2, it can be observed that for static frames the RGB modality provides better information, while for dynamic frames optical flow has less noisy information. This results in different semantic understanding across the individual modality networks.

Thus, it is desirable to design a collaborative framework that encourages the networks to learn a common understanding across different modalities for the same input scene. This way, if in a training iteration one of the networks cannot learn a proper representation for a certain region or time in its feature maps, it can use the knowledge from the other networks to improve its representations. An iterative occurrence of this event during the training process leads the networks to develop better representations in a collaborative manner.

Let $\mathbf{F}_m, \mathbf{F}_n \in \mathbb{R}^{W\times H\times T\times C}$ be the in-depth feature maps of two networks corresponding to the mth modality and nth modality, where W, H, T and C denote width, heights, the number of frames and channels of the feature maps, respectively. An in-depth feature map should contain high-level content representations (semantics) [219]. The element $\mathbf{f}^m_{i,j,t} \in \mathbb{R}^C$ in $\mathbf{F}_m$ represents the content for a certain block of time and spatial position. It is reasonable to expect the network m to develop correlated elements in $\mathbf{F}_m$ for spatiotemporal blocks with similar contents and semantics in the input. Thus, in an ideal case, the correlated elements in $\mathbf{F}_m$ should have correlated counterpart elements in $\mathbf{F}_n$.

The correlations between all the elements of $\mathbf{F}_m$ is expressed by its correlation matrix defined as follows

$$\mathbf{corr}(\mathbf{F}_m) = \hat{\mathbf{F}}_m \hat{\mathbf{F}}^T_m \in \mathbb{R}^{d\times d}, \tag{8.1}$$

where $\hat{\mathbf{F}}_m \in \mathbb{R}^{d\times C}$ contains the normalized elements of $\mathbf{F}_m$ in its rows, and $d = WHT$ is the number of elements in $\mathbf{F}^m$. The element $\mathbf{f}^m_{i,j,t}$ is normalized as $\hat{\mathbf{f}}^m_{i,j,t} = \tilde{\mathbf{f}}^m_{i,j,t}/\|\tilde{\mathbf{f}}^m_{i,j,t}\|$ where $\|\tilde{\mathbf{f}}^m_{i,j,t}\|$ is the magnitude of $\tilde{\mathbf{f}}^m_{i,j,t}$, and $\tilde{\mathbf{f}}^m_{i,j,t}$ is calculated by $\hat{\mathbf{f}}^m_{i,j,t} = \frac{\mathbf{f}^m_{i,j,t} - \mu_{i,j,t}}{\sigma_{i,j,t}}$, where $\mu_{i,j,t}$ and $\sigma_{i,j,t}$ are respectively the sample mean and variance of the element. We encourage the networks of the mth and the nth modalities to share a common correlation matrix for the feature maps of $\mathbf{F}_m$ and $\mathbf{F}_n$ so that they can have similar understanding for the input video while being free to have different styles. We do this by minimizing their *spatiotemporal semantic alignment* loss defined as

$$\ell^{m,n}_{SSA} = \rho^{m,n} \|\mathbf{corr}(\mathbf{F}_m) - \mathbf{corr}(\mathbf{F}_n)\|^2_F, \tag{8.2}$$

where $\rho^{m,n}$ is an adaptive regularization parameter defined in Section 3..2.

The *spatiotemporal semantic alignment* loss is closely related to the covariance matrix alignment of the source and target feature maps in domain adaptation methods [220, 221]. In addition, in some style transfer methods, the Gram matrices of feature maps are aligned [222, 219]. Aligning the Gram matrices, as opposed to our approach, discards the positional information and aligns the styles. In contrast, our method aligns the positional and temporal information and discards the style.

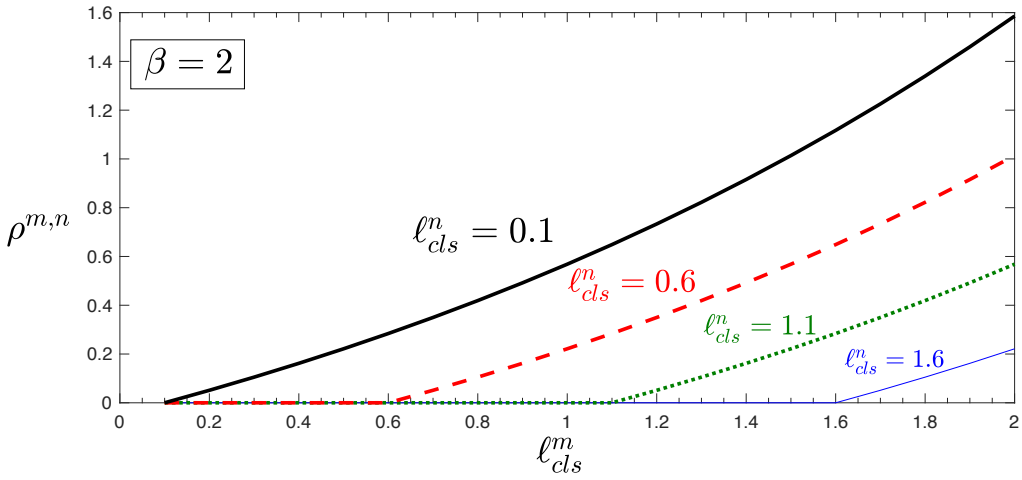


Figure 8.3: The value of focal regularization parameter ($\rho^{m,n}$) when $\beta = 2$ for different values of classification losses, ℓ^m_{cls} and ℓ^n_{cls}. Proportional to the classification performances of networks m and n, this parameter scales the SSA loss to focus on transferring positive knowledge.

3..2 Avoiding Negative Transfer

As discussed earlier, some modalities may provide weak features as compared to the others. In addition, even the strong modalities may sometimes have corrupted or hard examples in their training set. In these cases, aligning the spatiotemporal semantics of the representations from the other networks to the semantics of a week network may lead to a decrease in the performance. In such a case, a negative transfer has occurred. It is desirable to develop a method that produces positive knowledge transfer between the networks while avoiding negative transfer. Such a method in our framework should enforce the networks to only mimic the semantics of more accurate networks in learning the representations for their hard examples. To address this issue, we regularize our SSA loss with an adaptive regularization parameter termed as *focal regularization parameter*. This parameter is denoted as $\rho^{m,n}$ in equation (8.2).

In order to measure the performance of the network modalities, we can use their classification loss values. Assume ℓ^m_{cls} and ℓ^n_{cls} are the classification losses of the networks m and n that respectively correspond to the mth and nth modalities. In addition, let $\Delta\ell = \ell^m_{cls} - \ell^n_{cls}$ be their difference. A positive $\Delta\ell$ indicates that network n works better than network m. Hence, in training the network m, for large positive values of $\Delta\ell$, we want large values for $\rho^{m,n}$ to enforce the network to mimic the representations of the network n. As $\Delta\ell \rightarrow 0^+$, network n becomes

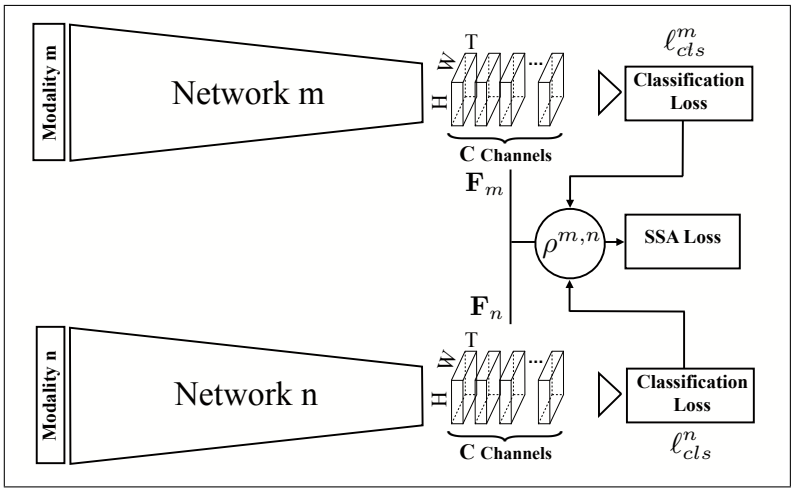


Figure 8.4: Training network m with the knowledge of network n. Training network m, is primarily done with respect to its classifier loss (ℓ^m_{cls}), but comparing with ℓ^n_{cls}, $\rho^{m,n}$ determines if involving the *SSA* loss is necessary, and if yes, it regularizes this loss with respect to the difference between the performances of two networks. Note that in the test time, both networks perform independently.

less an assist. Hence, we aim to have smaller $\rho^{m,n}$s to focus more on the classification task. Finally, negative $\Delta\ell$ indicates that the network n does not have better representations than the network m, and therefore $\rho^{m,n}$ should be zero to avoid the negative transfer. To address these properties, we define the *focal regularization parameter* as follows

$$\rho^{m,n} = \mathbf{S}(e^{\beta\Delta\ell} - 1) = \begin{cases} e^{\beta\Delta\ell} - 1 & \Delta\ell > 0 \\ 0 & \Delta\ell \leq 0 \end{cases} \tag{8.3}$$

where β is a positive focusing parameter, and $\mathbf{S}(\cdot)$ is the thresholding function at zero.

Figure 8.3 visualizes values of $\rho^{m,n}$ for various ℓ^n_{cls}s and $\ell^m_{cls} \in [0, 2]$, when $\beta = 2$. As can be seen, the parameter is dynamically scaled, where the scaling factor decays to zero as confidence in the classification performance of the current network modality increases (measured using $\Delta\ell$). This scaling factor can automatically down-weight the contribution of the shared knowledge if the performance of the modality network n is degraded (measured by ℓ^n_{cls}).

The focal regularization parameter $\rho^{m,n}$ is used as the regularization factor when aligning the correlation matrix of $\mathbf{F}^m$ in mth modality network to the correlation matrix of $\mathbf{F}^n$ in nth modality network.

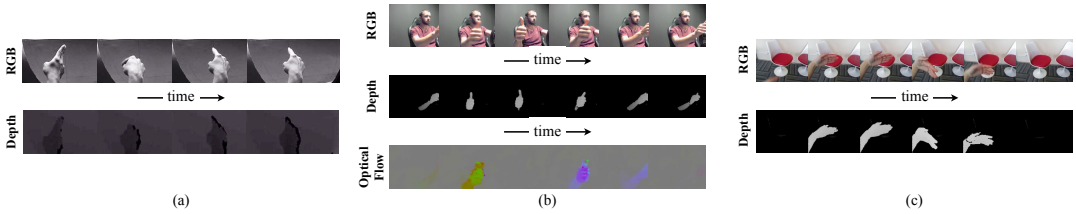


Figure 8.5: Sample sequences from different modalities of used datasets. (a) VIVA hand gesture dataset [1]. (b) NVGesture dataset [3]. (c) EgoGesture [2, 14]. As can be seen, the modalities in VIVA and EgoGesture datasets are well-aligned, while the depth map is not quite aligned with RGB and Optical flow maps in NVGesture.

3..3 Full Objective of the Modality Networks

Combining the aforementioned objectives, our full objective for training the network corresponding to the mth modality in an M-modality task is as follows

$$\ell^m = \ell^m_{cls} + \lambda \sum_{n=1}^{M} \ell^{m,n}_{SSA} \tag{8.4}$$

where λ is a positive regularization parameter. Note that for $n = m$, $\rho^{m,n} = 0$ and thus $\ell^{m,n}_{SSA} = 0$.

Figure 8.4 shows an overview of how the representations for the nth modality affects on learning the representation in the mth modality. Since $\rho^{m,n}$ is differentiable, the training can be done in an end-to-end manner.

Our model encourages the networks to improve their representation learning in the training stage. During testing, each network performs separately. Thus, once the networks are trained, one can use an individual modality network to acquire efficient recognition. However, it is worth mentioning that with our framework, applying a decision level modality fusion in the test stage is also possible. In fact, our experiments show that the proposed method not only improves the performance of unimodal networks, but it can also improve the fusion performance.

4. Experimental Results

In this section, we evaluate our method against state-of-the-art dynamic hand gesture methods. We conduct our experiments on three publicly available multimodal dynamic hand gesture datasets. The following datasets are used in our experiments.

- *VIVA hand gestures dataset* [1] is a multimodal dynamic hand gesture dataset specifically designed with difficult settings of cluttered background, volatile illumination, and frequent occlusion for studying natural human activities in real-world driving settings. This dataset was captured using a Microsoft Kinect device, and contains 885 visible RGB and depth video sequences (RGB-D) of 19 hand gesture classes, collected from 8 subjects.

- *EgoGesture dataset* [2, 14] is a large multimodal hand gesture dataset collected for the task of egocentric gesture recognition. This dataset contains 24,161 hand gesture clips of 83 classes of gestures, performed by 50 subjects. Videos in this dataset include both static and dynamic gestures captured with an Intel RealSense SR300 device in RGB-D modalities across multiple indoor and outdoor scenes.

- *NVGestures dataset* [3] has been captured with multiple sensors and from multiple viewpoints for studying human-computer interfaces. It contains 1532 dynamic hand gestures recorded from 20 subjects inside a car simulator with artificial lighting conditions. This dataset includes 25 classes of hand gestures. The gestures were recorded with SoftKinetic DS325 device as the RGB-D sensor and DUO-3D for the infrared streams. In addition, the optical flow and infrared disparity map modalities can be calculated from the RGB and infrared streams, respectively. We use RGB, depth and optical flow modalities in our experiments. Note that IR streams in this dataset do not share the same view with RGB, depth and optical flow modalities. The optical flow is calculated using the method presented in [223].

Figure 8.5 (a), (b), and (c) show sample frames from the different modalities of these datasets that are used in our experiments. Note that the RGB and depth modalities are well-aligned in the VIVA and EgoGesture datasets, but are not completely aligned in the NVGestures dataset.

For all the datasets, we compare our method against two state-of-the-art action recognition networks, I3D [15] and C3D [203], as well as state-of-the-art dynamic hand gesture recognition methods that were reported on the used datasets. In the tables, we report the results of our method as *"Multimodal Training Unimodal Testing" (MTUT)*.

Implementation Details: In the design of our method, we adopt the architecture of I3D network as the backbone network of our modality networks, and employ its suggested implementation details [15]. This network is an inflated version of Inception-V1 [224], which contains several 3D convolutional layers followed with 3D max-pooling layers and inflated Inception-V1 submodules. The detailed architecture can be found in [15]. We select the output of the last inflated Inception submodule, "`Mixed_5c`", as the in-depth feature map in our modality networks for applying the SSA loss (8.2). In all the experiments λ is set to 50×10^{-3}, and $\beta = 2$. The threshold function in the focal regularization parameter is implemented by a ReLu layer. For all the experiments with our method and I3D benchmarks, unless otherwise stated, we start with the publicly available ImageNet [174] + Kinetics [225] pre-trained networks.

We set the momentum to 0.9, and optimize the objective function with the standard SGD optimizer. We start with the base learning rate of 10^{-2} with a $10\times$ reduction when the loss is saturated. We use a batch size of 6 containing 64-frames snippets in the training stage. The models were implemented in Tensor-Flow 1.9 [128]. For our method, we start with a stage of pretraining with only applying the classification losses on the modality networks for 60 epochs, and then continue training with the SSA loss for another 15 epochs.

We employ the following spacial and temporal data augmentations during the training stage. For special augmentation, videos are resized to have the smaller video size of 256 pixels, and then randomly cropped with a 224×224 patch. In addition, the resulting video is randomly but consistently flipped horizontally. For temporal augmentation, 64 consecutive frames are picked randomly from the videos. Shorter videos are randomly padded with zero frames on both sides to obtain 64 frames. During testing, we use 224×224 center crops, apply the models convolutionally over the full video, and average predictions.

Note that we follow the above mentioned implementation details identically for the experiments with both the I3D method [15], and our method. The only difference between the I3D method and our MTUT is in their learning objective. In our case, it consists of the introduced constraints as well.

	Testing modality	
Method	RGB	Depth
HOG+HOG2 [1]	52.3	58.6
CNN:LRN [194]	57.0	65.0
C3D [203]	71.26	68.32
I3D [15]	78.25	74.46
MTUT (ours)	**81.33**	**81.31**

Table 8.1: 8-fold cross-subject average accuracies of different hand gesture methods on the VIVA hand gesture dataset [1]. The top performer is denoted by boldface.

4..1 VIVA Hand Gestures Dataset

In this set of experiments, we compare our method on the VIVA dataset against a hand-crafted approach (HOG+HOG2 [1]), a recurrent CNN-based method (CNN:LRN [194]), a C3D [203] model which were pre-trained on Sports-1M dataset [212] as well as the I3D method that currently holds the best results in action recognition [15]. All the results are reported by averaging the classification accuracies over 8-fold cross-subject cross-validation.

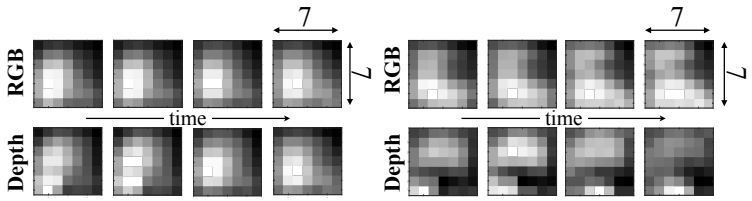
Table 8.1 shows the performance of the dynamic hand gesture methods tested on the visible and depth modalities of the VIVA dataset. As can be seen from this table, the I3D network performs significantly better than *HOG+HOG2* and *CNN:LRN*. This is in part due to the knowledge that I3D contains from its pretraining on ImageNet and Kinematic datasets. Nonetheless, we observe that our method that shares the same architecture and settings with the I3D networks and only differs in the learning procedure has significantly improved the I3D method by a 3.08% boost in the performance of RGB's network and 6.85% improvement on the performance of the depth's network. This experiment shows that our method is able to integrate the complementary information between two different modalities to learn efficient representations that can improve their individual performances.

4..2 EgoGesture Dataset

We assess the performance of our method along with various hand gesture recognition methods published on the large-scale hand gesture dataset, EgoGesture [2]. Table 8.2 compares unimodal test accuracies of different hand gesture methods. VGG16 [175] is a frame-based

Method	Testing modality	
	RGB	Depth
VGG16 [175]	62.5	62.3
VGG16 + LSTM [92]	74.7	77.7
C3D [203]	86.4	88.1
C3D+LSTM+RSTTM [2]	89.3	90.6
I3D [15]	90.33	89.47
MTUT (ours)	**92.48**	**91.96**

Table 8.2: Accuracies of different hand gesture methods on the EgoGesture dataset [2]. The top performer is denoted by boldface.

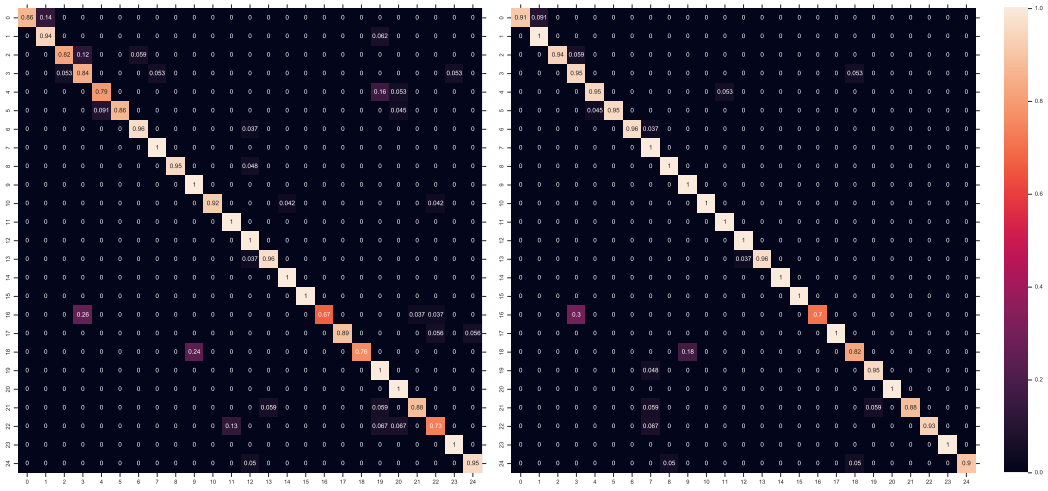


(a) MTUT (ours) (b) I3D

Figure 8.6: Visualization of the feature maps corresponding to the layer "`Mixed_5c`" in different networks for a sample input from EgoGesture dataset. These figures show the sequence of average feature maps (over 1024 channels) in (a) the RGB and depth networks trained with the I3D method. (b) the RGB and depth networks trained with our method. Intensity displays the magnitude.

recognition method, and VGG16+LSTM [92] combines this method with a recurrent architecture to leverage the temporal information as well. As can be seen, the 3D-CNN-based methods, C3D, C3D+LSTM+RSTMM [2], and I3D, outperform the VGG16-based methods. However, among the 3D-CNN-based methods, our method outperforms the top performers in the RGB domain by 2.15% and in the Depth domain by 1.09%.

In Figure 8.6, we visualize a set of feature maps from the RGB and depth networks trained with the I3D and our method. We feed a given input from the EgoGesture dataset to different networks and calculate the average of feature maps over the channels in the layer "`Mixed_5c`". We display the resulting sequence in four 7×7 blocks. Here the temporal dimension is four and the spatial content is 7×7. Layer "`Mixed_5c`" is the layer in the I3D architecture in which we apply the SSA loss to. We observe that the networks trained with our model have learned to detect similar structures for the given input (Figure 8.6 (a)). On the other hand, the



(a) (b)

Figure 8.7: The confusion matrices obtained by comparing the grand-truth labels and the predicted labels from the RGB network trained on the NVGesture dataset by (a) I3D [15] model, and (b) our model. Best seen on the computer, in color and zoomed in.

networks trained with the I3D model are not bounded to develop similar structures. Thus, even though the input of the two modalities represent the same content, the feature maps may detect different structures (Figure 8.6 (b)).

4..3 NVGesture Dataset

In order to test our method on tasks with more than two modalities, in this section, we report the classification results on the RGB, depth and optical flow modalities of the NVGesture dataset [3]. The RGB and optical flow modalities are well-aligned in this dataset, however, the depth map includes a larger field of view (see Figure 8.5 (b)).

Table 8.3 tabulates the results of our method in comparison with the recent state-of-the-art methods: HOG+HOG2, improved dense trajectories (iDT) [226], R3DCNN [3], two-stream CNNs [204], and C3D as well as human labeling accuracy. The iDT [226] method is often recognized as the best performing hand-crafted method [227]. However, we observe that similar to the previous experiments the 3D-CNN-based methods outperform the other hand gesture recognition methods, and among them, our method provides the top performance in all the modalities. This table confirms that our method can improve the unimodal test performance by leveraging the knowledge from multiple modalities in the training stage. This is despite the

	Testing modality		
Method	RGB	Depth	Opt. Flow
HOG+HOG2 [1]	24.5	36.3	-
Two Stream CNNs [204]	54.6	-	68.0
C3D [203]	69.3	78.8	-
iDT [226]	59.1	-	76.8
R3DCNN [3]	74.1	80.3	77.8
I3D [15]	78.42	82.28	83.19
MTUT (ours)	**81.33**	**84.85**	**83.40**
Human labeling accuracy:		88.4	

Table 8.3: Accuracies of different unimodal hand gesture methods on the NVGesture dataset [3]. The top performer is denoted by boldface.

fact that the depth map in this dataset is not completely aligned with the RGB and optical flow maps.

Figure 8.7 evaluates the coherence between the predicted labels and ground-truths in our method and compares it with I3D for the RGB modality of the NVGesture dataset. This coherence is calculated by their confusion matrices. We observe that our method has less confusion between the input classes and provides generally a more diagonalized confusion matrix. This improvement is better observed in the first six classes.

4..4 Effect of Unimodal Improvements on Multimodal Fusion

As previously discussed, our method is designed for embedding knowledge from multiple modalities in unimodal networks for improving their unimodal test performance. In this section, we examine if the enhanced unimodal networks trained by our approach can also improve the accuracy of a decision level fusion that is calculated from the average of unimodal predictions. The decision level fusion of different modality streams is currently the most common fusion technique in the top performer dynamic action recognition methods [15, 203, 204].

In Table 8.4 and Table 8.5 we compare the multimodal-fusion versions of our method (MTUT^{F}) to state-of-the-art multimodal hand gesture recognition systems tested on the VIVA hand gesture and EgoGesture datasets, respectively. As can be seen, our method shows the top multimodal fusion performance on both datasets. These tables show that if multiple modalities

Method	Fused modalities	Accuracy
HOG+HOG2 [1]	RGB+Depth	64.5
CNN:LRN [194]	RGB+Depth	74.4
CNN:LRN:HRN [194]	RGB+Depth	77.5
C3D [203]	RGB+Depth	77.4
I3D [15]	RGB+Depth	83.10
MTUT^{F} (ours)	RGB+Depth	**86.08**

Table 8.4: Accuracies of different multimodal fusion-based hand gesture methods on the VIVA dataset [1]. The top performer is denoted by boldface.

Method	Fused modalities	Accuracy
VGG16 [175]	RGB+Depth	66.5
VGG16 + LSTM [92]	RGB+Depth	81.4
C3D [203]	RGB+Depth	89.7
C3D+LSTM+RSTTM [2]	RGB+Depth	92.2
I3D [15]	RGB+Depth	92.78
MTUT^{F} (ours)	RGB+Depth	**93.87**

Table 8.5: Accuracies of different multimodal fusion hand gesture methods on the EgoGesture dataset [3]. The top performer is denoted by boldface.

are available at the test time, the improved performance of unimodal networks gained by training with our model can also result in an improved multimodal fusion performance in the test time.

Similarity, in Table 8.6 we report the multimodal fusion results on the NVGesture dataset. Note that since this dataset includes three modalities, based on the modalities we include in the training stage, we report multiple versions of our method. We report the version of our method that includes all three modalities in the training stage as $\text{MTUT}^{\text{F}}_{\text{all}}$, and the versions that only involve (RGB+Depth) and (RGB+Optical-Flow) in their training as $\text{MTUT}^{\text{F}}_{\text{RGB-D}}$ and $\text{MTUT}^{\text{F}}_{\text{RGB-OF}}$, respectively. While all versions of our method outperform the other multimodal fusion methods in Table 8.6, the performances of $\text{MTUT}^{\text{F}}_{\text{RGB-D}}$ and $\text{MTUT}^{\text{F}}_{\text{all}}$ in the fusion of RGB+Depth is worth highlighting. $\text{MTUT}^{\text{F}}_{\text{all}}$ in this experiment has also been trained on the absent modality, the optical flow, while $\text{MTUT}^{\text{F}}_{\text{RGB-D}}$ has been only trained on the RGB and Depth modalities. We observe that $\text{MTUT}^{\text{F}}_{\text{all}}$ has successfully integrated the knowledge of the absent modality and provided a better performance at the test time.

Method	Fused modalities	Accuracy
HOG+HOG2	RGB+Depth	36.9
I3D [15]	RGB+Depth	83.82
$\text{MTUT}^{\text{F}}_{\text{RGB-D}}$ (ours)	RGB+Depth	85.48
$\text{MTUT}^{\text{F}}_{\text{all}}$ (ours)	RGB+Depth	**86.10**
Two Stream CNNs [204]	RGB+Opt. flow	65.6
iDT [226]	RGB+Opt. flow	73.4
I3D [15]	RGB+Opt. flow	84.43
$\text{MTUT}^{\text{F}}_{\text{RGB-OF}}$ (ours)	RGB+Opt. flow	**85.48**
$\text{MTUT}^{\text{F}}_{\text{all}}$ (ours)	RGB+Opt. flow	**85.48**
R3DCNN [3]	RGB+Depth+Opt. flow	83.8
I3D [15]	RGB+Depth+Opt. flow	85.68
$\text{MTUT}^{\text{F}}_{\text{all}}$ (ours)	RGB+Depth+Opt. flow	**86.93**
Human labeling accuracy:		88.4

Table 8.6: Accuracies of different multimodal fusion hand gesture methods on the NVGesture dataset [2]. The top performer is denoted by boldface.

4..5 Analysis of the Network

To understand the effects of some of our model choices, we explore the performance of some variations of our model on the VIVA dataset. In particular, we compare our method with and without the *focal regularization parameter* and the *SSA* loss. Beside our I3D-based method, we analyze these variations on a different backbone network, C3D [203] as well. C3D is another recently proposed activity recognition architecture. We name this method MTUT_{C3D}. Besides, we use C3D+SSA and I3D+SSA to refer to versions of our method with C3D and I3D backbones that contain a variation of the *SSA* loss that does not have the *focal regularization parameter*. For MTUT_{C3D} and C3D+SSA, we apply the SSA loss on feature maps of the last maxpooling layer ("`MaxPool3d_5`").

To provide a fair comparison setting, we train these networks from scratch on the VIVA dataset, and report their performances in Table 8.7. As can be seen, the top performer is our I3D-based network with both SSA and focal regularization parameter. Several interesting observations can be made from the results in Table 8.7. As the table reveals, the I3D-based methods generally perform better than the C3D-based methods. This coincides with the previous reports [15]. In addition, C3D+SSA and I3D+SSA methods in the case of RGB networks show improvements and in the case of depth modality have comparable results as compared to their

	Testing modality	
Method	RGB	Depth
C3D	53.05	55.65
C3D+SSA	53.73	54.52
MTUT_{C3D}	**56.56**	**58.71**
I3D	65.72	67.30
I3D+SSA	65.83	66.96
MTUT	**68.43**	**71.26**

Table 8.7: Comparison of variations of MTUT with C3D and I3D backbones trained from scratch.

base networks C3D and I3D, respectively. However, the top performers in both modalities are the full version of our method applied on these base networks. This clearly shows the importance of our focal regularization parameter in avoiding negative transfer when transferring the knowledge between the modalities. Note that C3D, I3D and MTUT are trained from scratch in this experiment, while in the Table 8.1 we reported their performance on the networks trained with pre-trained weights.

5. Conclusion

We presented a new framework to leverage the knowledge of multiple modalities when training unimodal networks that can independently work at the test time inference with improved accuracy. Our model trains separate 3D-CNNs per available modalities, and shares their knowledge by the introduced *spatiotemporal semantic alignment* loss. We also regularized this loss with a *focal regularization parameter* that ensures that only positive knowledge is transferred between the modality networks, and negative transfer is avoided. Our experiments confirmed that our method can provide remarkable improvements to the unimodal networks at the test time. We also showed that the enhanced unimodal networks that are trained with our method can contribute to an improved multimodal fusion performance at test time as well.

The incorporation of our method for multimodal learning in other applications is a topic of further research.

Chapter 9

Multimodal Categorization of Crisis Events in Social Media

1. Introduction

Each second, billions of images and texts that capture a wide range of events happening around us are uploaded to social media platforms from all over the world. At the same time, the fields of Computer Vision (CV) and Natural Language Processing (NLP) are rapidly advancing [228, 229, 230] and are being deployed at scale. With large-scale visual recognition and textual understanding available as fundamental tools, it is now possible to identify and classify events across the world in real-time. This is possible, to some extent, in images and text separately, and in limited cases, using a combination. A major difficulty in crisis events,[1] in particular, is that as events surface and evolve, users post fragmented, sometimes conflicting information in the form of image-text pairs. This makes the automatic identification of notable events significantly more challenging.

Unfortunately, in the middle of a crisis, the information that is valuable for first responders and the general public often comes in the form of image-text pairs. So while traditional CV and NLP methods that treat visual and textual information separately can help, a big gap exists in current approaches. Despite the general consensus on the importance of using AI for Social Good [231, 232, 233], the power of social media, and a long history of interdisciplinary research on humanitarian crisis efforts, there has been very little work on automatically detecting crisis events *jointly* using visual and textual information.

Prior approaches that tackle the detection of crisis events have focused on either image-only or text-only approaches. As shown in Figure 9.1, however, an image alone can be ambiguous in terms of its urgency whereas the text alone may lack details.

[1] An event that is going (or is expected) to lead to an unstable and dangerous situation affecting an individual, group, community, or whole society (from Wikipedia); typically requiring an emergency response.

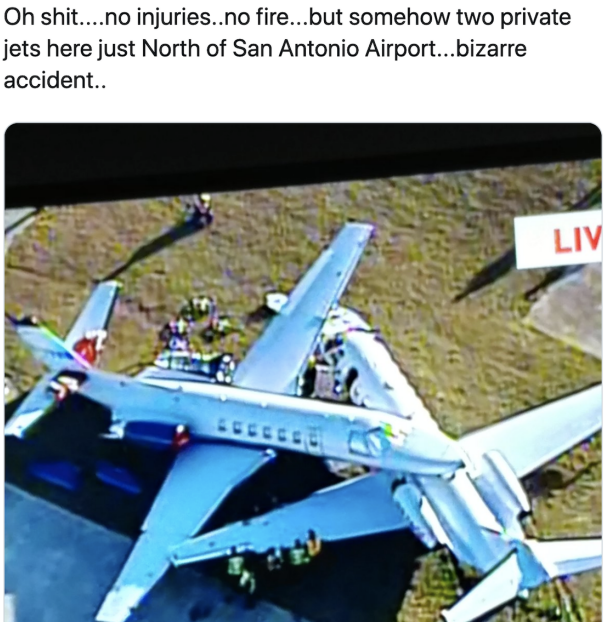


Figure 9.1: A crisis-related image-text pair from social media

To address these issues, we propose a framework to detect crisis events using a combination of image and text information. In particular, we present an approach to automatically label images, text, and image-text pairs based on the following criteria/tasks: 1) **Informativeness**: whether the social media post is useful for providing humanitarian aid in an emergency event, 2) **Event Classification**: identifying the type of emergency (in Figure 9.2, we show some of the categories that different image-text pairs belong to in our event classification task), and 3) **Severity**: rating how severe the emergency is based on the damage indicated in the image and text. Our framework consists of several steps in which, given an image-text pair, we create a feature map for the image, generate word embeddings for the text, and propose a cross-attention mechanism to fuse information from the two modalities. It differs from previous multimodal classification in how it deals with fusing that information.

In short, we present a novel, multimodal framework for classification of multimodal data in the crisis domain. This approach, ”Cross Attention”, avoids transferring negative knowledge between modalities and makes use of stochastic shared embeddings to mitigate overfitting in small data as well as dealing with training data with inconsistent labels for different modalities. Our model outperforms strong unimodal and multimodal baselines by up to 3 F-score points across three crisis tasks.

Figure 9.2: Samples from Task 2; Event Classification with Texts and Images.

2. Related Work

AI for Emergency Response: Recent years have seen an explosion in the use of Artificial Intelligence for Social Good [231, 232, 233]. Social media has proven to be one of most relevant and diverse resources and testbeds, whether it be for identifying risky mental states of users [234, 235, 236], recognizing emergent health hazards [237], filtering for and detecting natural disasters [238, 239, 240], or surfacing violence and aggression in social media [241].

Most prior work on detecting crisis events in social media has focused on text signals. For instance, Kumar *et al.* [242] propose a real-time tweet-tracking system to help first responders gain situational awareness once a disaster happens. Shekhar *et al.* [243] introduce a crisis analysis system to estimate the damage level of properties and the distress level of victims. At a large scale, filtering (e.g., by anomaly or burst detection), identifying (e.g., by clustering), and categorizing (e.g., by classifying) disaster-related texts on social media have been the foci of multiple research groups [244, 245, 246], achieving accuracy levels topping at 0.75 on small annotated datasets collected from Twitter.

Disaster detection in images has been an active front, whether it be user-generated content or satellite images (for a detailed survey, refer to Said *et al.* [238]). For instance, Ahmad *et*

al. [247] introduce a pipeline method to effectively link remote sensor data with social media to better assess damage and obtain detailed information about a disaster. Li *et al.* [248] use convolutional neural networks and visualization methods to locate and quantify damage in a disaster images. Nalluru *et al.* [249] combine semantic textual and image features to classify the relevancy of social media posts in emergency situations.

Our framework focuses on combining images and text, yielding performance improvements on three disaster classification tasks.

Deep Multimodal Learning: In deep multimodal learning, neural networks are used to integrate the complementary information from multiple representations (modalities) of the same phenomena [250, 120, 251, 252]. In many applications, including image captioning [253, 254], visual question answering [91, 255], and text-image matching [256, 257, 258], combining image and text signals is of interest. Thus many recent works study image-text fusion [259, 260, 261, 262].

Existing multimodal learning frameworks applied to the crisis domain are relatively limited. Lan *et al.* [263] combine early fusion and late fusion methods to incorporate their advantages, Ilyas [264] introduce a disaster classification system based on naive-bayes classifiers and support vector machines. Kelly *et al.* [265] introduce a system for real-time extraction of information from text and image content in Twitter messages with exploiting the spatio-temporal metadata for filtering, visualizing, and monitoring flooding events. Mouzannar *et al.* [266] propose a multimodal deep learning framework to identify damage related information on social media posts with texts, images, and video.

In the application of crisis tweets categorization, one modality may contain uninformative or even misleading information. The attention module in our model passes information based on the confidence in the usefulness of different modalities. The more confident modality blocks weak or misleading features from the other modality through their cross-attention link. The partially blocked results of both modalities are later judged by a self-attention layer to decide which information should be passed to the next layer. While our attention module is closely related to co-attention and self-attention mechanisms [267, 268, 269, 255, 270, 254], unlike

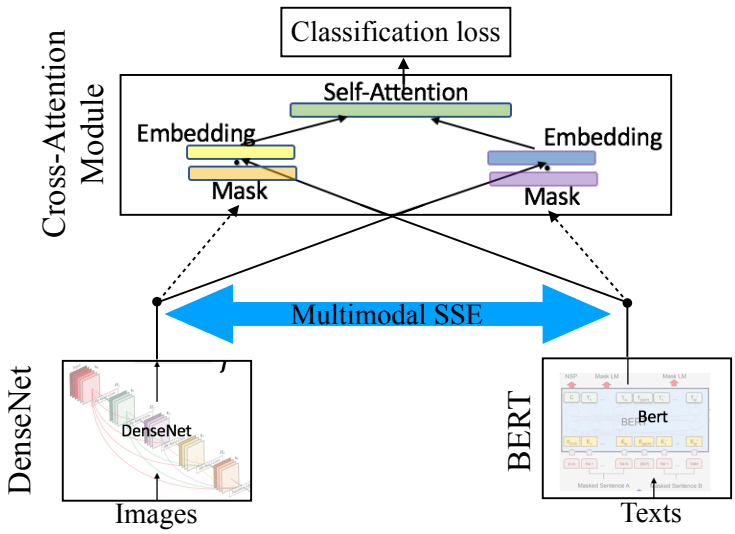


Figure 9.3: Illustration of Our Framework. Embedding features are extracted from images and texts by DenseNet and BERT networks, respectively, and are integrated by the cross-attention module. In the training process, the embeddings of different samples are stochastically transitioned between each other to provide a robust regularization.

them, it does not need the input features to be homogeneous. In contrast, self-attention and co-attention layers can be sensitive to heterogeneous inputs. The details of the model are described in the next section.

3. Methodology

The architecture we propose is designed for classification problems that takes as input image-text pairs such as user generated tweets in social media, as illustrated in Figure 9.3, where the DenseNet and BERT graphs are from [178] and [230]. Our methodology consists of 4 parts: the first two parts extract feature maps from the image and extract embeddings from the text, respectively; the third part comprises our cross-attention approach to fuse projected image and text embeddings; and the fourth part uses Stochastic Shared Embeddings (SSE) [271] as our regularization technique to prevent over-fitting and deal with training data with inconsistent labels for image and text pairs.

We describe each module in the sub-sections that follow.

3..1 Image Model for Feature Map Extraction:

We extract feature maps from images using Convolutional Neural Networks (CNNs). In our model we select DenseNet [178], which reduces module sizes and increases connections between layers to address parameter redundancy and improve accuracy (other approaches, such as EfficientNet [272] could also be used, but DenseNet is efficient and commonly used for this task).

For each image v_i, we therefore have:

$$f_i = \textbf{DenseNet}(v_i), \tag{9.1}$$

where v_i is the input image, $f_i \in \mathbb{R}^{D_f}$ is the vectorized form of a deep feature map in the DenseNet with dimension $D_f = W \times H \times C$, where W, H, C are the feature map's height, width and number of channels respectively.

3..2 Text Model for Embedding Extraction:

Full-network pre-training [273, 230] has led to a series of breakthroughs in language representation learning. Specifically, deep-bidirectional Transformer models such as BERT [230] and its variants [274, 275] have achieved state-of-the-art results on various natural language processing tasks by leveraging close and next-sentence prediction tasks as weakly-supervised pre-training. Therefore, we use BERT as our core model for extracting embeddings from text (variants such as XLNET [274] and ALBERT [275] could also be used). We use the BERT model pre-trained on Wiki and Books data[276] on crisis-related tweets t_i's. For each text input t_i, we have

$$e_i = \textbf{BERT}(t_i), \tag{9.2}$$

where t_i is a sequence of word-piece tokens and $e_i \in \mathbb{R}^{756}$ is the sentence embedding. Similar to the BERT paper [230], we take the embedding associated with [CLS] to represent the whole sentence.

In the next subsection we detail how DenseNet and BERT are fused.

3..3 Cross-attention module for avoiding negative knowledge in fusion:

After we obtain the image feature map f_i (DenseNet) and the sentence embedding e_i (BERT), we use a new cross-attention mechanism to fuse the information they represent. In many text-vision tasks, the input pair can contain noise. In particular, in classification of tweets, one modality may contain non-informative or even misleading information. In such a case, negative information transfer can occur. Our model can mitigate the effects of one modality over another on a case by case basis.

To address this issue, in our cross-attention module, we use a combination of cross-attention layers and a self-attention layer. In this module, each modality can block the features of the other modality based on its confidence in the usefulness of its input. This happens with the cross-attention layer. The result of partially blocked features from both modalities is later fed to a self-attention layer to decide which information should be passed to the next layer.

The self-attention layer exploits a fully-connected layer to project the image feature map into a fixed dimensionality K (we use $K = 100$), and similarly project the sentence embedding so that:

$$\begin{aligned} \tilde{f_i} &= F(W_v^T f_i + b_v), \\ \tilde{e_i} &= F(W_e^T e_i + b_e), \end{aligned} \tag{9.3}$$

where F represents an activation function such as ReLU (used in our experiments) and both $\tilde{f_i}$ and $\tilde{e_i}$ are of dimension $K = 100$.

In the case of misleading information in one modality, without an attention mechanism (such as co-attention [259]), the resulting $\tilde{f_i}$ and $\tilde{e_i}$ cannot be easily combined without hurting performance. Here, we propose a new attention mechanism called cross-attention (Figure 9.3), which differs from standard co-attention mechanisms: the attention mask α_{v_i} for the image is completely dependent on the text embedding e_i, while the attention mask α_{e_i} for the text is completely dependent on the image embedding f_i. Mathematically, this can be expressed as follows:

$$\begin{aligned} \alpha_{v_i} &= \sigma(W_v'^T f_i + b_v'), \\ \alpha_{e_i} &= \sigma(W_e'^T e_i + b_e'), \end{aligned} \tag{9.4}$$

where σ is the Sigmoid function. Co-attention, in contrast, can be expressed as follows:

$$\alpha_{v_i} = \sigma(W_v'^T[f_i|e_i] + b_v'),$$
$$\alpha_{e_i} = \sigma(W_e'^T[f_i|e_i] + b_e'), \tag{9.5}$$

where | means concatenation.

After we have the attention masks $\alpha_{v_i}, \alpha_{e_i}$ for image and text respectively, we can augment the projected image and text embeddings $\tilde{f_i}, \tilde{e_i}$ with $\alpha_{v_i} \cdot \tilde{f_i}$ and $\alpha_{e_i} \cdot \tilde{e_i}$ before performing concatenation or adding. In our experiments, we use concatenation but obtained similar performance using addition.

The last step of this module takes the concatenated embedding which jointly represents the image and text tuple in and feeds into the two-layer fully-connected networks. We add self-attention in the fully-connected networks and use the standard softmax cross-entropy loss for the classification.

In Section 4., we show that the combination of cross-attention layers and the self-attention layer on their concatenation works better than co-attention and self-attention mechanisms for the tasks we address in this chapter.

3..4 SSE for Better Regularization

Due to unforeseeable and unpredictable nature of disasters, and also because they require fast processing and reaction, one often has to deal with limited annotations for user-generated content during crises. Using regularization techniques to mitigate this issue becomes especially important. In this section, we extend Stochastic Shared Embeddings (SSE) technique [271] to its multimodal version for taking the full advantage from the annotated data by 1) generating new artificial multimodal pairs. 2) also including the annotated data with inconsistent labels for text and image in the training process.

SSE-Graph [271], a variation of SSE, is a data-driven approach for regularizing embedding layers which uses a knowledge graph to stochastically make transitions between embeddings of different samples during the stochastic gradient descent (SGD). That means, during the training, based on a knowledge graph, there is a chance that embeddings of different samples being swapped. We use the text and image labels to construct knowledge graphs that can be used

to create stochastic multimodal training samples with consistent labels for both the image and text.

We treat feature maps of images as embeddings and use class labels to construct knowledge graphs. The feature maps of two images are connected by an edge in the graph, if and only if they belong to the same class (e.g. they are both labeled "affected individuals"). We follow the same procedure for text embeddings and construct a knowledge graph for text embeddings as well. Finally, we connect the nodes associated with the knowledge graph of image feature maps with an edge to nodes in text's knowledge graph if and only if they belong to the same class.

Let Φ^v and Φ^t be sets of parameters. We define the transition probability $p(i^v, j^v|\Phi^v)$ as probability of transition from i^v to j^v, where i^v and j^v are nodes in the image knowledge graph that correspond to image features f_i and f_j. Similarly, we define $p(i^t, k^t|\Phi^t)$ as probability of transition from i^t to k^t (nodes corresponding to text embeddings e_i and e_k, respectively).

Taking image feature maps as an example, if i^v is connected to j^v but not connected to l^v in the knowledge graph, one simple and effective way to generate more multimodal pairs is to use a random walk (with random restart and self-loops) on the knowledge graph. Since we are more interested in transitions within embeddings of consistent labels, in each transition probability, we set the ratio of $p(i^v, j^v|\Phi^v)$ and $p(i^v, l^v|\Phi^v)$ to be a constant greater than 1. In more formal notation, we have

$$i^v \sim j^v, i^v \nsim l^v \longrightarrow p(i^v, j^v|\Phi^v)/p(i^v, l^v|\Phi^v) = \rho^v, \tag{9.6}$$

where ρ^v is a tuning parameter and $\rho^v > 1$, and $\sim$ and $\nsim$ denote connected and not connected nodes in the knowledge graph. We also have:

$$p(i^v, i^v|\Phi) = 1 - p_0^v, \tag{9.7}$$

where p_0^v is called the *SSE probability* for image features.

We similarly define ρ^t and p_0^t in $\Phi^t = \{\rho^t, p_0^t\}$ for text embeddings. Note that ρ^t is defined with respect to the image features' label. That is

$$i^v \sim j^t, i^v \nsim l^t \longrightarrow p(i^t, j^t|\Phi^t)/p(i^t, l^t|\Phi^t) = \rho^t. \tag{9.8}$$

Both Φ^v and Φ^t parameters sets are treated as tuning hyper-parameters in experiments and can be tuned fairly easily. With Eq. (9.8), Eq. (9.7) and $\sum_{k^v} p(j^v, k^v|\Phi^v), \sum_{k^t} p(j^t, k^t|\Phi^t) = 1$, we can derive transition probabilities between any two sets of feature maps in images and texts to fill out the transition probability table.

With the right parameter selection, each multimodal pair in the training can be transitioned to many more multimodal pairs that are highly likely to have consistent labels for the image and text pairs which can mitigate both the issues of limited number of training samples and inconsistency in the annotations of image-text pairs.

4. Experimental Setup

The image-text classification problem we consider can be formulated as follows: we have as input $(v_1, t_1), \ldots, (v_i, t_i), \ldots, (v_n, t_n)$, where n is the number of training tuples and the i-th tuple consists of both image v_i and text t_i. The respective labels for v_i and t_i's are also given in training data. Our goal is to predict the correct label for any unseen (v, t) pair. To simplify the evaluation, we assume there is only one correct label associated with the unseen (v, t) pairs. As a result, this chapter targets a multi-class classification problem instead of a multi-label problem.

4..1 Dataset

There are very few crisis datasets, and to the best of our knowledge there is only one *multimodal* crisis dataset, CrisisMMD [277]. It consists of annotated image-tweet pairs where images and tweets are independently labeled as described below. We use this dataset for our experiments. The dataset was collected using event-specific keywords and hashtags during seven natural disasters in 2017: Hurricane Irma, Hurricane Harvey, Hurricane Maria, the Mexico earthquake, California wildfires, Iraq-Iran earthquakes, and Sri Lanka floods. The corpus is comprised of three types of manual annotations:

Task 1: Informative vs. Not Informative: whether a given tweet text or image is useful for humanitarian aid purposes, defined as providing assistance to people in need.

Task 2: Humanitarian Categories: given an image, or tweet, or both, categorize it into one of the five following categories:

- Infrastructure and utility damage
- Vehicle damage
- Rescue, volunteering, or donation efforts
- Affected individuals (injury, dead, missing, found, etc.)
- Other relevant information

Note that we merge the data that are labeled as *injured or dead people* and *missing or found people* in the CrisisMMD with those that are labeled as *affected individuals* and view all of them as one class of data.

Task 3: Damage Severity: assess the severity of damage reported in a tweet image and classify it into Severe, Mild, and Little/None.

It is important to note that while the annotations for the last task are only on images. Our experiments reveal that using tweet texts along with the images can boost performance. In addition, our method is the first one to perform all three tasks on this dataset (text-only, image-only, combined).

4..2 Settings

Images and text from tweets in this dataset were annotated independently. Thus, in many cases, images and text in the same pairs may not share the same labels for either Task 1 or Task 2 (labels for Task 3 were only created by annotating the images). Given the different evaluation conditions, we carry out three evaluation settings for the sake of being comprehensive in our model assessment but also to establish best practices for the community: *Setting A*: we exclude the image-text pairs with differing labels for image and text; *Setting B*: we include the image-text pairs with different labels in the training set but keep the test set the same as in A.

Table 9.1: Number of samples in different splits of our settings.

Setting	# of Training samples	# of Dev samples	# of Test samples
Setting A			
Task1:	7876	553	2821
Task2:	1352	540	1467
Task3:	2590	340	358
Setting B			
Task1:	12680	553	2821
Task2:	5433	540	1467
Setting C			
Experiment 1:	174	-	217
Experiment 2:	4037	-	217
Experiment 3:	4761	-	217

Table 9.2: Setting A: Informativeness Task, Humanitarian Categorization Task and Damage Severity Task Evaluations.

	Informativeness Task			Humanitarian Categorization Task			Damage Severity Task		
Model	Acc	Macro F1	Weighted F1	Acc	Macro F1	Weighted F1	Acc	Macro F1	Weighted F1
DenseNet [178]	81.57	79.12	81.22	83.44	60.45	86.96	62.85	52.34	66.10
BERT [230]	84.90	81.19	83.30	86.09	66.83	87.83	68.16	45.04	61.09
Compact Bilinear Pooling[255]	88.12	86.18	87.61	89.30	67.18	90.33	66.48	**61.03**	**70.58**
Compact Bilinear Gated Pooling [61]	88.76	87.50	88.80	85.34	65.95	89.42	68.72	51.46	65.34
MMBT [278]	82.48	81.27	82.15	85.82	64.78	88.66	65.36	52.12	69.34
Score Fusion	88.16	83.46	85.26	86.98	54.01	88.96	71.23	53.48	66.26
Feature Fusion	87.56	85.20	86.55	89.17	67.28	91.40	67.60	40.62	56.47
Attention Variant 1 (Ours)	89.29	85.68	87.04	88.41	64.60	90.71	71.51	55.41	69.71
Attention Variant 2 (Ours)	88.34	86.12	87.42	89.23	67.63	91.56	63.13	58.03	69.39
Attention Variant 3 (Ours)	88.20	86.22	87.47	87.18	64.67	90.24	68.99	57.42	69.16
SSE-Cross-BERT-DenseNet (Ours)	**89.33**	**88.09**	**89.35**	**91.14**	**68.41**	**91.82**	**72.65**	59.76	70.41

In addition, we introduce *Setting C* to mimic a realistic crisis tweet classification task where we only train on events that have transpired before the event(s) in the test set.

Table 9.1 shows the number of samples in each set for different setting and tasks.

Setting A: In this setting our train and test data is sampled from tweets in which the text and image pairs have the same label. That is:

$$C(v_i) = C(t_i), \tag{9.9}$$

where $C(x)$ denotes the class of data point x. This results in a small, yet potentially more reliable training set. We mix the data from all seven crisis events and split the data into training, dev and test sets.

Setting B: We relax the assumption in Equation 9.9 and allow in training:

$$C(v_i) \neq C(t_i), \tag{9.10}$$

As the training set of this setting contains samples with inconsistent labels for image and text, multimodal fusion methods such as late feature fusion cannot deal with the training data. Our method, on the other hand, with the use of the proposed multimodal SSE, can transition the training instance with in consistent labels to a new training pair with consistent labels. We do this by manually setting $p_0^t = 1$ for the training cases with inconsistent image-text labels (i.e. all the text samples are transitioned). Since unimodal models only receive one of the modalities, it is also possible to train them separately on images and texts and use an average of their prediction in the testing stage (also known as score level fusion).

However, we maintain the assumption of Eq. (9.9) for the test data. This helps to directly compare the two settings with the same test samples. In fact, in practice, the data is most valuable when the class labels match for both image and text. The rationale is that detecting an event is more valuable to crisis managers than the categorization of different parts of that event. Our dev and test sets for this setting are similar to the previous setting. However, the training set contains a larger number of samples where their image-text pairs are not necessarily labeled as the same class.

Setting C: This setting is closest to the real-world scenario where we analyze the new event of a crisis with a model trained on previous crisis events. First, we require the training and test sets to be from crisis events of a different nature (i.e., wildfire vs. flood). Second, we maintain the temporal component and only train on events that have happened before the tweets of the testing set. Since collecting annotated data on an urgent ongoing-event is not possible, and also because an event of crisis may do not have a similar annotated event in the past, these two restrictions often simulate a real-world scenario. For the experiments of this setting, there is no dev set. Instead, we use a random portion of the training data to tune the hyper-parameters.

We test on the tweets that are related to the California Wildfire (Oct 10 - 27, 2017), and train on the following three sets:

1. Sri Lanka Floods tweets (*May 31- Jul. 3, 2017*)
2. Sri Lanka Floods, and Hurricane Harvey and Hurricane Irma tweets (*May 31- Sept. 21 , 2017*)
3. Sri Lanka Floods, Hurricanes Harvey and Irma and Mexico Earthquakes (*May 31 - Oct.*

5, 2017).

Similar to setting B, for the test set (i.e. California Wildfire) we only consider the samples with consistent labels for image and text, but for the training sets, we use all the available samples.

4..3 Baselines

We compare our method against several state-of-the-art methods for text and/or image classification. There are a number of categories of baseline methods we compare against. In the first category, we compare to DenseNet and BERT, which are of the most comonnly used unimodal classification networks for images and texts respectively. We use Wikipedia pre-trained BERT and pre-trained DenseNet on ImageNet [156], and fine-tune them on the training sets.

The second category of baseline methods include several recently proposed multimodal fusion methods for classification:

- Compact Bilinear Pooling [255]: multimodal compact bilinear pooling is a fusion technique first used in visual question answering task but can be easily modified to perform standard classification task.
- Compact Bilinear Gated Pooling [61]: this fusion method is an adaptation of the compact bilinear pooling method where an extra attention gate is added on top the compact bilinear pooling module.
- MMBT [278]: recently proposed supervised multimodal bitransformers model for classifying images and text.

The third category is the score level *Score Fusion* and late feature fusion *Feature Fusion* of DenseNet and BERT networks. Score level fusion is one of the most common fusion techniques. It averages the predictions of separate networks trained on the different modalities. Feature Fusion is one of the most effective methods for integrating two modalities [122]. It concatenates deep layers from modality networks to predict a shared output. We also provide three variations of our attention modules and report their performance: The first variant is to replace cross-attention of Eq. (9.4) with co-attention of Eq. (9.5); the second variant is to remove self-attention; the third variant is to change the cross-attention with self-attention modules.

We compare our model, SSE-Cross-BERT-DenseNet, to the baseline models above.

4..4 Evaluation Metrics

We evaluate the models in this chapter using classification accuracy,[2] Macro F1-score and weighted F1-score. Note that while in the event of a crisis, the number of samples from different categories often significantly varies, it is important to detect all of them. F1-score and weighted F1-score take both false positives and false negatives into account, and therefore, along with accuracy as an intuitive measure, are proper evaluation metrics for our datasets.

4..5 Training Details

We use pre-trained DenseNet and BERT as our image and text backbone networks, and fine-tune them separately on text-only and image-only training samples. The details of their implementations can be found in [178] and [230], respectively. We do not freeze the pre-trained weights and train all the layers for both the backbone networks.

We use the standard SGD optimizer. We start with the base learning rate of 2×10^{-3} with a $10\times$ reduction when the dev loss is saturated. We use a batch size of 32. The models were implemented in Keras and Tensorflow-1.4 [128]. In all the applicable experiments, we select hyper-parameters with cross-validation on the accuracy of dev set. For the experiments in Setting 3 that we do not have an evaluation set, we tune hyper-parameters on 15% of the training samples. We select ρ^v, ρ^t and p_0^v, p_0^t respectively in the range of $\rho^v, \rho^t \in [10, 20000]$ and $p_0^v, p_0^t \in [0, 1]$.

We employ the following data augmentations on the images during the training stage. Images are resized such that the smallest side is 228 pixels, and then randomly cropped with a 224×224 patch. In addition, we produce more images by randomly flipping the resulting image horizontally.

For tweet normalization, we remove double spaces and lower case all characters. In addition, we replace any hyperlink in the tweet with the sentinel word "`link`".

---

[2] In the settings that our experiments are defined classification accuracy is equivalent to Micro F1-score.

Table 9.3: Setting B: Informativeness Task and Humanitarian Categorization Task Evaluations

Model	Informativeness Task			Humanitarian Categorization Task		
	Accuracy	Macro F1	Weighted F1	Accuracy	Macro F1	Weighted F1
DenseNet [178]	83.36	80.95	82.95	82.89	66.68	83.13
BERT [230]	86.26	84.44	86.01	87.73	83.72	87.57
Score Fusion	87.03	85.19	86.90	91.41	83.26	91.36
SSE-Cross-BERT-DenseNet (Ours)	**90.05**	**88.88**	**89.90**	**93.46**	**84.16**	**93.35**
Best from Table 9.2	89.33	88.09	89.35	91.48	67.87	91.34

Table 9.4: Comparing our proposed method with baselines for Humanitarian Categorization Task in Setting 3. We fix the last occurred crisis namely 'California wildfires' as test data and vary the training data which is specified in the columns.

Model	Sri Lanka Floods			Sri Lanka Floods + Hurricanes Harvey & Irma			Sri Lanka Floods + Hurricanes Harvey & Irma + Mexico earthquake		
	Accuracy	Macro F1	Weighted F1	Accuracy	Macro F1	Weighted F1	Accuracy	Macro F1	Weighted F1
DenseNet [178]	55.71	35.77	56.85	70.32	52.23	68.55	70.32	44.80	68.79
BERT [230]	31.96	20.90	27.21	73.97	53.90	73.51	74.43	56.98	74.21
Score Fusion	56.62	36.77	57.96	81.74	56.54	81.03	81.28	55.90	80.54
SSE-Cross-BERT-DenseNet (Ours)	**62.56**	**39.82**	**62.08**	**84.02**	**63.12**	**83.55**	**86.30**	**65.55**	**85.93**

5. Experimental Results

5..1 Setting A: Excluding The Training Pairs with Inconsistent Labels

As shown in Table 9.2, our proposed framework, SSE-Cross-BERT-DenseNet, easily outperforms the standalone DenseNet and BERT models. Compared with baseline methods Compact Bilinear Pooling [255], Compact Bilinear Gated Pooling [61], and MMBT [278], our proposed cross-attention fusion method does enjoy an edge over previous known fusion methods, including the standard score fusion and feature fusion. This edge holds true across Settings A, B and C. In section 5..4, we conduct an ablation study to investigate which components (SSE, cross-attention, and self-attention) have the most impact on model performance.

One important observation we find across the three tasks is that despite the fact that accuracy percentages are reasonably good for simple feature fusion method, the macro F1 scores improve much more once we add attention mechanisms.

5..2 Setting B: Including The Training Pairs with Inconsistent Labels

In this setting, we investigate whether our models can perform better if we can make use of more labelled data for un-matched images and texts. Note that this involves training on noisier data than the prior setting. In Table 9.3, our proposed framework SSE-Cross-BERT-DenseNet

Table 9.5: Ablation Study of our proposed method for Humanitarian Categorization Task in Setting A.

Model	Test Set		
	Accuracy	Macro F1	Weighted F1
SSE-Cross-BERT-DenseNet (Ours)	**91.14**	**68.41**	**91.82**
− Self-Attention	89.23	56.50	87.70
− Cross-Attention	88.48	56.38	87.10
− Cross-Attention + Co-Attention	88.41	64.60	90.71
− Cross-Attention + Self-Attention	86.30	58.33	85.27
− Dropout	83.37	54.83	82.46
− SSE	88.41	64.60	90.71
− SSE + Shuffling Within Class	88.68	62.91	88.33
− SSE + Mix-up [279]	89.16	54.63	87.37

beats the best results from Setting A for both the Informativeness Task (89.90 to 89.35 Weighted F1) and the Humanitarian Categorization Task (93.35 to 91.34). The gap between our method versus standalone BERT and DenseNet also widens. Note that the test sets are the same for setting A and setting B while only the training data differs.

5..3 Setting C: Temporal

This setting is designed to resemble a realistic scenario where the available data is (1) only from the past (i.e. the train / test sets are split in the order they occurred in the real world). (2) train and test sets are not from the same crisis. We find that our proposed model consistently performs better than standalone image and text models (see Table 9.4). Additionally, performance increases for all models, including ours, with the inclusion of more crisis data to train on. This emphasizes the importance of collecting and labelling more crisis data even if there is no guarantee that the crises we collected data from will be similar to a future one. In the experiments, training crises contain floods, hurricanes and earthquakes but the test crisis is fixed at wildfires.

5..4 Ablation Study

In our ablation study, we examine each component of the model in Figure 9.3: namely self-attention on concatenated embedding, cross-attention on fusing image feature map & sentence embedding, dropout and SSE regularization. All the experiments in this section are conducted in Setting A. First, we find self-attention plays an important role on the final performance,

accuracy drops to 89.23 from 91.14 if self-attention is removed. Second, the choice of cross-attention over co-attention and self-attention is well justified: we see the accuracy performance drops to around 88 by replacing the cross-attention. Third, dropout regularization [280] plays an important role in regularizing the hidden units: if we remove dropout completely, performance suffers a large drop from 91.14 to 83.37. Fourthly, we justify the usage of SSE [271] over the choice of Mixup [279] or within-class shuffling data augmentation. SSE performs better than mixup in terms of accuracy 91.14% versus 89.16%, and even much better in terms of F1 scores, 68.41 versus 54.63 for macro F1 score and 91.82 versus 87.37 for weighted F1 score.

6. Conclusions and Future Work

In this chapter, we presented a novel multimodal framework for fusing image and textual inputs. We introduced a new cross attention module that can filter not-informative or misleading information from modalities and only fuse the useful information. We also presented a multimodal version of Stochastic Shared Embeddings (SSE) to regularize the training process and deal with limited training data. We evaluate this approach on three crisis tasks involving social media posts with images and text captions. We show that our approach not only outperforms image-only and text-only approaches which have been the mainstay in the field, but also other multimodal combination approaches.

For future work we plan to test how our approach generalizes to other multimodal problems such as sarcasm detection in social media posts [281, 282], as well as experiment with different image and text feature extractors. Given that the CrisisMMD corpus is the only dataset available for this task and it is limited in size, we also aim to construct a larger set, which is a major effort.

Chapter 10

Conclusion and Future Work

1. Conclusion

In this thesis, we focused on approaches for learning from massive high-dimensional and multimodal data. We mentioned that high-dimensional data often confuse algorithms with irrelevant dimensions, noises, and the curse of dimensionality. We aimed to develop multimodal algorithms that can leverage the complementary information from multimodal and high-dimensional data.

We started off focusing on *subspace clustering* algorithms that are widely popular for representing high-dimensional data in low dimension feature spaces. In **Chapter 3**, we extended the popular sparse and low-rank based subspace clustering methods to multimodal subspace clustering algorithms that can integrate multiple high-dimensional modalities and represent them in low-dimensional common subspaces. Then, in **Chapter 4**, we used CNNs to improve our proposed multimodal subspace clustering methods and developed deep multimodal subspace clustering networks. In **Chapter 5**, we showed that these subspace clustering networks could benefit from data augmentation techniques. We introduced a framework to incorporate data augmentation techniques in subspace clustering networks.

In the second part of the thesis, we employed the findings of Chapters 3,4, and 5 and developed several multimodal classification approaches. **Chapter 6** borrowed the idea of deep subspace clustering networks and applied it to the task of sparse representation-based classification (SRC). We showed that when the training data is limited and the data is very high-dimensional, deep sparse representation-based classification (DSRC) performs better than state-of-the-art neural network architectures. **Chapter 7** extended our DSRC to its multimodal version.

Chapter 8 and **Chapter 9** focused on two real-world applications of high-dimensional

multimodal data. **Chapter 8** argued using multiple video streams such as calculating optical-flow as a new modality in dynamic hand gesture recognition challenges the state-of-the-art methods in performing in real-time. Thus, we introduced a novel method to train unimodal networks with knowledge from multimodal inputs. The unimodal networks that are trained with our method leveraged the knowledge of multiple modalities and performed in real-time at the test time. **Chapter 9** introduced a fusion method for combining the information in texts and images of social media posts. Both texts and images are considered high-dimensional data, and in the case of social media posts, they can sometimes be uninformative or even misleading. The method that we presented in **Chapter 9** is able to prevent the negative knowledge from the inputs to be fused in the joint representation that is learned from the text-image pairs.

2. Future Work

2..1 Subspace Clustering of Heterogeneous Data

In a multimodal learning task, it is assumed that the multiple modalities (presentations) are bundled together. Thus, for example in a two modality task, per each presentation, there is a paired representation in the other modality. However, if paired representations are not available across different modalities we face a new task. In this case, we are given a collection of data from multiple modalities (domains), and we aim to segment the heterogeneous data based on their class labels.

In many applications, one has to deal with heterogeneous [1] data. For example, when clustering digits, one may have to process both computer generated as well as handwritten digits. Similarly, when clustering face images collected in the wild, one may have to cluster images of the same individual collected using different cameras and possibly under different resolution and lighting conditions. Clustering of heterogeneous data is difficult because it is not meaningful to directly compare the heterogeneous samples with different distributions which may span different feature spaces. In recent years, various domain adaptation methods have been developed to deal with the distributional changes that occur after learning a classifier for supervised and semi-supervised learning [283]. However, these methods have not been developed

[1] Data with different sizes or different natures.

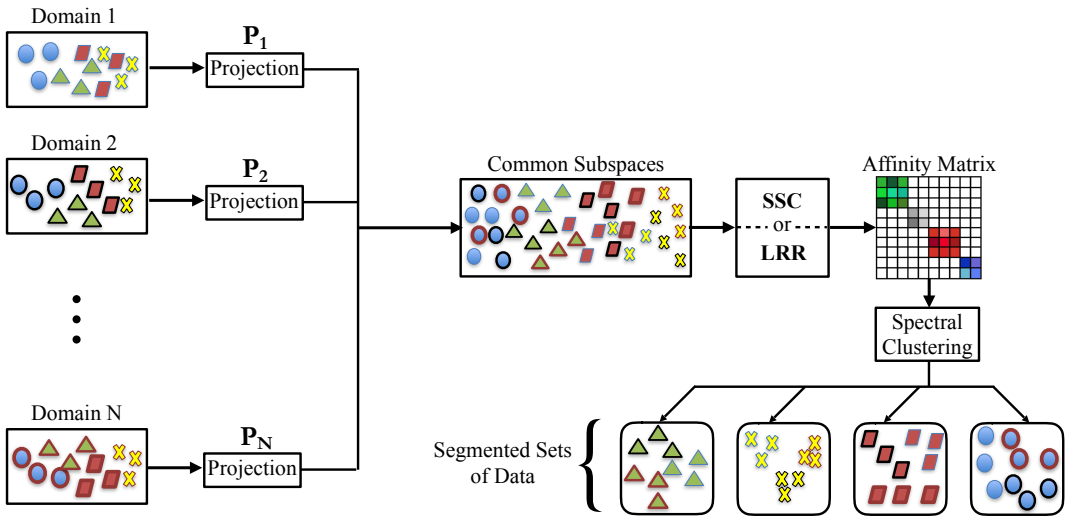


Figure 10.1: An overview of the proposed for subspace clustering of heterogeneous data.

for clustering heterogeneous data that lie in a union of low-dimensional subspaces.

Hence, we propose domain adaptive versions of the sparse and low-rank subspace clustering methods. Figure 10.1 gives an overview of the proposed method. Given data from K different domains, we simultaneously learn the projections and find the sparse or low-rank representation in the projected common subspace. Once the projection matrices and the sparse or low-rank coefficient matrix is found, it can be used for subspace clustering.

2..2 Adversarial Domain Adaptive Subspace Clustering

In biometrics recognition, one is often faced with a challenge of matching biometric samples that are collected under different environmental conditions. For example, in face recognition one may have to match a well-lit face image with an image that is acquired in a poor illumination condition. Another issue that what we often face in biometrics recognition is the problem of cross-sensor matching, where the test samples are verified using data enrolled with a different sensor. As new sensors are being developed for acquiring the biometric samples and existing ones are being upgraded, this becomes an important issue. Regardless of the cause of the domain shift, any distributional change (i.e. environmental, cross-sensor change, resolution etc.) that occurs after learning a classifier can degrade its performance at test time. Various domain adaptation techniques have been developed in the literature to mitigate this degradation

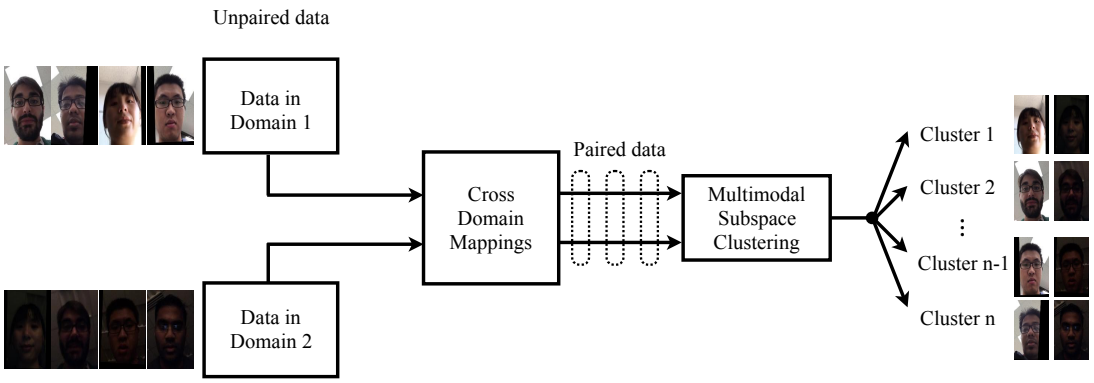


Figure 10.2: An overview of the proposed adversarial domain adaptive subspace clustering framework.

[283].

The domain adaptation problem can be defined in many different ways including semi-supervised domain adaptation [284, 285, 286, 287] and unsupervised domain adaptation [288, 289, 290, 291, 292]. In semi-supervised domain adaptation, both source and target domains are assumed to have partial labels. In contrast, in unsupervised domain adaptation, only the source domain is assumed to have partial labels and the target domain is assumed to be completely unlabeled. Another important domain adaptation problem that is often encountered in practice but is not widely studied in the literature is the problem of domain adaptive clustering, where no label information is assumed to be known [136]. This is particularly a difficult problem because we have no side information such as labels to group the samples from the same class from different domains in a single cluster.

Thus, we propose a new method for domain adaptive subspace clustering in which we use adversarial networks to approximate the mapping functions that map the source data into the target domain and the target data into the source domain. Using these mapping functions, we map the available data to their counter domains and obtain a paired representation of the data corresponding to different domains. Once the paired representation of the data is obtained, we exploit their self expressiveness property and employ multimodal sparse and low-rank subspace clustering methods [49] to cluster the paired representations with respect to their subspaces. Figure 10.2 gives an overview of the proposed adversarial domain adaptive (ADA) subspace clustering framework.

2..3 Sarcasm Detection and Other Multimodal Applications in Social Media Posts

Chapters 8 and 9 of this thesis focused on developing approaches for two use cases of multimodal learning in real-world applications. Dynamic hand gesture recognition and categorization of crisis events. One possible direction for future work is focusing on the deployment of the proposed methods in other multimodal applications. Sarcasm detection in social media is one of these applications [281, 282]. Often the sarcasm in a post can only be detected by the contradiction or the especial semantic relation between the image and the textual input. This pattern motivates us to invest in designing special fusion techniques that can understand and compare the relations of contents across different modalities. A combination of the proposed methods in chapters 8 and 9 can be useful for this application. The semantic alignment loss in chapter 8 can be a metric for comparing the information across different modalities. The cross-attention module in chapter 9 can learn to use these relations for detecting sarcasm.

One other possible path is to approach this problem with graph neural networks. Graph neural networks can extract the relations between different entities. One use graph neural networks to combine information from an external knowledge graph with the input pairs from social media to provide more

References

[1] E. Ohn-Bar, M. M. Trivedi, Hand gesture recognition in real time for automotive interfaces: A multimodal vision-based approach and evaluations, IEEE Transactions on Intelligent Transportation Systems 15 (6) (2014) 2368–2377.

[2] C. Cao, Y. Zhang, Y. Wu, H. Lu, J. Cheng, Egocentric gesture recognition using recurrent 3d convolutional neural networks with spatiotemporal transformer modules, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2017, pp. 3763–3771.

[3] P. Molchanov, X. Yang, S. Gupta, K. Kim, S. Tyree, J. Kautz, Online detection and classification of dynamic hand gestures with recurrent 3d convolutional neural network, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2016, pp. 4207–4215.

[4] P. Ji, T. Zhang, H. Lia, M. Salzmann, I. Reid, Deep subspace clustering networks, in: Advances in Neural Information Processing Systems (NeurIPS), 2017.

[5] Y. LeCun, C. Cortes, Mnist handwritten digit database, AT&T Labs [Online]. Available: http://yann. lecun. com/exdb/mnist (2010).

[6] J. J. Hull, A database for handwritten text recognition research, IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI) 16 (5) (1994) 550–554.

[7] S. Hu, N. J. Short, B. S. Riggan, C. Gordon, K. P. Gurton, M. Thielke, P. Gurram, 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.

[8] K. C. Lee, J. Ho, D. J. Kriegman, Acquiring linear subspaces for face recognition under variable lighting, IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI) 27 (5) (2005) 684–698.

[9] S. A. Nene, S. K. Nayar, H. Murase, et al., Columbia object image library (coil-20) (1996).

[10] S. A. Nene, S. K. Nayar, H. Murase, Columbia object image library (coil-100) (1996).

[11] F. S. Samaria, A. C. Harter, Parameterisation of a stochastic model for human face identification, in: Proceedings of 1994 IEEE Workshop on Applications of Computer Vision, IEEE, 1994, pp. 138–142.

[12] Y. Netzer, T. Wang, A. Coates, A. Bissacco, B. Wu, A. Y. Ng, Reading digits in natural images with unsupervised feature learning, in: NIPS workshop on deep learning and unsupervised feature learning, Vol. 2011, 2011, p. 5.

[13] H. Zhang, V. M. Patel, S. Shekhar, R. Chellappa, Domain adaptive sparse representation-based classification, in: Automatic Face and Gesture Recognition (FG), 2015 11th IEEE International Conference and Workshops on, Vol. 1, IEEE, 2015, pp. 1–8.

[14] Y. Zhang, C. Cao, J. Cheng, H. Lu, Egogesture: A new dataset and benchmark for egocentric hand gesture recognition, IEEE Transactions on Multimedia 20 (5) (2018) 1038–1050.

[15] J. Carreira, A. Zisserman, Quo vadis, action recognition? a new model and the kinetics dataset, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), IEEE, 2017, pp. 4724–4733.

[16] R. Basri, D. W. Jacobs, Lambertian reflectance and linear subspaces, IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI) 25 (2) (2003) 218–233.

[17] T. Hastie, P. Y. Simard, Metrics and models for handwritten character recognition, Statistical Science (1998) 54–65.

[18] J. P. Costeira, T. Kanade, A multibody factorization method for independently moving objects, International Journal of Computer Vision 29 (3) (1998) 159–179.

[19] R. Vidal, Subspace clustering, IEEE Signal Processing Magazine 28 (2) (2011) 52–68. doi:10.1109/MSP.2010.939739.

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

[21] A. Y. Yang, J. Wright, Y. Ma, S. S. Sastry, Unsupervised segmentation of natural images via lossy data compression, Computer Vision and Image Understanding 110 (2) (2008) 212–225.

[22] W. Hong, J. Wright, K. Huang, Y. Ma, Multiscale hybrid linear models for lossy image representation, IEEE Transactions on Image Processing 15 (12) (2006) 3655–3671.

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

[24] R. Vidal, Y. Ma, S. Sastry, Generalized principal component analysis (gpca), IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI) 27 (12) (2005) 1–15.

[25] T. E. Boult, L. G. Brown, Factorization-based segmentation of motions, in: IEEE Workshop on Visual Motion, 1991, pp. 179–186.

[26] J. Ho, M. H. Yang, J. Lim, K. Lee, D. Kriegman, Clustering appearances of objects under varying illumination conditions, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2003.

[27] T. Zhang, A. Szlam, G. Lerman, Median k-flats for hybrid linear modeling with many outliers, in: Workshop on Subspace Methods, 2009.

[28] S. Rao, R. Tron, R. Vidal, Y. Ma, Motion segmentation in the presence of outlying, incomplete, or corrupted trajectories, IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI) 32 (10) (2010) 1832–1845.

[29] H. Derksen, Y. Ma, W. Hong, J. Wright, Segmentation of multivariate mixed data via lossy coding and compression, in: SPIE Visual Communications and Image Processing, Vol. 6508, 2007.

[30] A. Y. Yang, S. R. Rao, Y. Ma, Robust statistical estimation and segmentation of multiple subspaces, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2006, pp. 99–99.

[31] A. Goh, R. Vidal, Segmenting motions of different types by unsupervised manifold clustering, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2007.

[32] J. Yan, M. Pollefeys, A general framework for motion segmentation: Independent, articulated, rigid, non-rigid, degenerate and non-degenerate, in: Proceedings of the European Conference on Computer Vision (ECCV), 2006.

[33] E. Elhamifar, R. Vidal, Sparse subspace clustering, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2009, pp. 2790–2797.

[34] G. Liu, Z. Lin, Y. Yu, Robust subspace segmentation by low-rank representation, in: International Conference on Machine Learning (ICML), 2010.

[35] P. Favaro, R. Vidal, A. Ravichandran, A closed form solution to robust subspace estimation and clustering, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2011.

[36] E. Elhamifar, R. Vidal, Sparse subspace clustering: Algorithm, theory, and applications, IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI) 35 (11) (2013) 2765–2781.

[37] G. Liu, Z. Lin, S. Yan, J. Sun, Y. Yu, Y. Ma, Robust recovery of subspace structures by low-rank representation, IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI) 35 (1) (2013) 171–184.

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

[39] V. M. Patel, H. V. Nguyen, R. Vidal, Latent space sparse and low-rank subspace clustering, IEEE Journal of Selected Topics in Signal Processing 9 (4) (2015) 691–701.

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

[41] K. Chaudhuri, S. M. Kakade, K. Livescu, K. Sridharan, Multi-view clustering via canonical correlation analysis, in: International Conference on Machine Learning (ICML), 2009, pp. 129–136.

[42] A. Kumar, P. Rai, H. Daume, Co-regularized multi-view spectral clustering, in: Advances in Neural Information Processing Systems (NeurIPS), 2011, pp. 1413–1421.

[43] X. Zhao, N. Evans, J. L. Dugelay, A subspace co-training framework for multi-view clustering, Pattern Recognition Letters 41 (2014) 73–82.

[44] K. Zhan, C. Zhang, J. Guan, J. Wang, Graph learning for multiview clustering, IEEE Transactions on Cybernetics (2018) 1–9doi:10.1109/TCYB.2017.2751646.

[45] M. White, X. Zhang, D. Schuurmans, Y. liang Yu, Convex multi-view subspace learning, in: Advances in Neural Information Processing Systems (NeurIPS), 2012, pp. 1673–1681.

[46] H. Wang, C. Weng, J. Yuan, Multi-feature spectral clustering with minimax optimization, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2014, pp. 4106–4113.

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

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

[49] M. Abavisani, V. M. Patel, Multimodal sparse and low-rank subspace clustering, Information Fusion 39 (2018) 168–177.

[50] Y.-X. Wang, H. Xu, C. Leng, Provable subspace clustering: When lrr meets ssc, in: Advances in Neural Information Processing Systems (NeurIPS), 2013, pp. 64–72.

[51] M. Abavisani, V. M. Patel, Deep multimodal subspace clustering networks, IEEE Journal of Selected Topics in Signal Processing 12 (6) (2018) 1601–1614.

[52] M. Abavisani, A. Naghizadeh, D. Metaxas, V. Patel, Deep subspace clustering with data augmentation, in: Advances in Neural Information Processing Systems (NeurIPS), 2020.

[53] M. Abavisani, A. Naghizadeh, D. N. Metaxas, V. M. Patel, Supplementary materials: Deep subspace clustering with data augmentation.

[54] M. Abavisani, V. M. Patel, Deep sparse representation-based classification, IEEE Signal Processing Letters 26 (6) (2019) 948–952.

[55] M. Abavisani, V. M. Patel, Deep multimodal sparse representation-based classification, in: 2020 IEEE International Conference on Image Processing (ICIP), 2020, pp. 773–777. doi:10.1109/ICIP40778.2020.9191317.

[56] M. Abavisani, H. R. V. Joze, V. M. Patel, Improving the performance of unimodal dynamic hand-gesture recognition with multimodal training, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2019.

[57] H. V. Joze, M. Abavisani, Video recognition using multiple modalities, uS Patent App. 16/287,113 (May 7 2020).

[58] M. Abavisani, L. Wu, S. Hu, J. Tetreault, A. Jaimes, Multimodal categorization of crisis events in social media, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2020, pp. 14679–14689.

[59] M. Abavisani, L. Wu, S. Hu, J. Tetreault, A. Jaimes, Supplementary materials: Multimodal categorization of crisis events in social media, 2020.

[60] S. Boyd, N. Parikh, E. Chu, B. Peleato, J. Eckstein, Distributed optimization and statistical learning via the alternating direction method of multipliers, Foundations and Trends in Machine Learning 3 (1) (2011) 1–122.

[61] D. Kiela, E. Grave, A. Joulin, T. Mikolov, Efficient large-scale multi-modal classification, arXiv preprint arXiv:1802.02892 (2018).

[62] C. Feichtenhofer, A. Pinz, A. Zisserman, Convolutional two-stream network fusion for video action recognition, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), IEEE, 2016, pp. 1933–1941.

[63] Y. Chen, C.-G. Li, C. You, Stochastic sparse subspace clustering, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2020, pp. 4155–4164.

[64] J. Zhang, C.-G. Li, C. You, X. Qi, H. Zhang, J. Guo, Z. Lin, Self-supervised convolutional subspace clustering network, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2019, pp. 5473–5482.

[65] P. Zhou, Y. Hou, J. Feng, Deep adversarial subspace clustering, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2018, pp. 1596–1604.

[66] M. Kheirandishfard, F. Zohrizadeh, F. Kamangar, Multi-level representation learning for deep subspace clustering, arXiv preprint arXiv:2001.08533 (2020).

[67] A. Y. Ng, M. I. Jordan, Y. Weiss, On spectral clustering: Analysis and an algorithm, in: Advances in Neural Information Processing Systems (NeurIPS), Vol. 2, 2002, pp. 849–856.

[68] M. Soltanolkotabi, E. Elhamifar, E. J. Candes, et al., Robust subspace clustering, The Annals of Statistics 42 (2) (2014) 669–699.

[69] U. Von Luxburg, A tutorial on spectral clustering, Statistics and computing 17 (4) (2007) 395–416.

[70] E. J. Candes, T. Tao, Decoding by linear programming, IEEE transactions on information theory 51 (12) (2005) 4203–4215.

[71] D. L. Donoho, For most large underdetermined systems of linear equations the minimal ??1-norm solution is also the sparsest solution, Communications on Pure and Applied Mathematics: A Journal Issued by the Courant Institute of Mathematical Sciences 59 (6) (2006) 797–829.

[72] S. Bickel, T. Scheffer, Multi-view clustering., in: IEEE International Conference on Data Mining, Vol. 4, 2004, pp. 19–26.

[73] C. Zhang, H. Fu, S. Liu, G. Liu, X. Cao, Low-rank tensor constrained multiview subspace clustering, in: IEEE International Conference on Computer Vision (ICCV), IEEE, 2015, pp. 1582–1590.

[74] X. Cao, C. Zhang, H. Fu, S. Liu, H. Zhang, Diversity-induced multi-view subspace clustering, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2015, pp. 586–594.

[75] X. Cao, C. Zhang, C. Zhou, H. Fu, H. Foroosh, Constrained multi-view video face clustering, IEEE Transactions on Image Processing 24 (11) (2015) 4381–4393.

[76] P. Wu, S. C. Hoi, H. Xia, P. Zhao, D. Wang, C. Miao, Online multimodal deep similarity learning with application to image retrieval, in: ACM international conference on Multimedia, ACM, 2013, pp. 153–162.

[77] R. Kiros, K. Popuri, D. Cobzas, M. Jagersand, Stacked multiscale feature learning for domain independent medical image segmentation, in: International Workshop on Machine Learning in Medical Imaging, Springer, 2014, pp. 25–32.

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

[79] 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 62 (4) (2015) 1132–1140.

[80] P. Mamoshina, A. Vieira, E. Putin, A. Zhavoronkov, Applications of deep learning in biomedicine, Molecular pharmaceutics 13 (5) (2016) 1445–1454.

[81] N. Neverova, C. Wolf, G. Taylor, F. Nebout, Moddrop: adaptive multi-modal gesture recognition, IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI) 38 (8) (2016) 1692–1706.

[82] S. S. Mukherjee, N. M. Robertson, Deep head pose: Gaze-direction estimation in multimodal video, IEEE Transactions on Multimedia 17 (11) (2015) 2094–2107.

[83] C. Ding, D. Tao, Robust face recognition via multimodal deep face representation, IEEE Transactions on Multimedia 17 (11) (2015) 2049–2058.

[84] 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 10 (2) (2016) 99–111.

[85] A. Jain, J. Tompson, Y. LeCun, 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.

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

[87] A. Naghizadeh, B. Razeghi, I. Radmanesh, M. Hatamian, R. E. Atani, Z. N. Norudi, Counter attack to free-riders: Filling a security hole in bittorrent protocol, in: 2015 IEEE 12th International Conference on Networking, Sensing and Control, IEEE, 2015, pp. 128–133.

[88] D. Gollmann, Computer security, Wiley Interdisciplinary Reviews: Computational Statistics 2 (5) (2010) 544–554.

[89] A. Naghizadeh, B. Razeghi, E. Meamari, M. Hatamian, R. E. Atani, C-trust: A trust management system to improve fairness on circular p2p networks, Peer-to-Peer Networking and Applications 9 (6) (2016) 1128–1144.

[90] B. Razeghi, M. Hatamian, A. Naghizadeh, S. Sabeti, G. A. Hodtani, A novel relay selection scheme for multi-user cooperation communications using fuzzy logic, in: 2015 IEEE 12th International Conference on Networking, Sensing and Control, IEEE, 2015, pp. 241–246.

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

[92] J. Donahue, L. Anne Hendricks, S. Guadarrama, M. Rohrbach, S. Venugopalan, K. Saenko, T. Darrell, Long-term recurrent convolutional networks for visual recognition and description, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2015, pp. 2625–2634.

[93] A. Karpathy, A. Joulin, L. F. Fei-Fei, Deep fragment embeddings for bidirectional image sentence mapping, in: Advances in Neural Information Processing Systems (NeurIPS), 2014, pp. 1889–1897.

[94] O. Vinyals, A. Toshev, S. Bengio, D. Erhan, Show and tell: A neural image caption generator, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), IEEE, 2015, pp. 3156–3164.

[95] M. Ren, R. Kiros, R. Zemel, Exploring models and data for image question answering, in: Advances in Neural Information Processing Systems (NeurIPS), 2015, pp. 2953–2961.

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

[97] J. Yang, J. Wright, T. S. Huang, Y. Ma, Image super-resolution via sparse representation, IEEE Transactions on Image Processing 19 (11) (2010) 2861–2873.

[98] H. Zhang, V. M. Patel, R. Chellappa, Multitask multivariate common sparse representations for robust multimodal biometrics recognition, in: IEEE International Conference on Image Processing, 2015, pp. 202–206.

[99] H. Zhang, V. M. Patel, R. Chellappa, Robust multimodal recognition via multitask multivariate low-rank representations, in: IEEE International Conference and Workshops on Automatic Face and Gesture Recognition, 2015, pp. 1–8.

[100] M. R. Hestenes, Multiplier and gradient methods, Journal of optimization theory and applications 4 (5) (1969) 303–320.

[101] V. Simoncini, Computational methods for linear matrix equations, SIAM Review 58 (3) (2016) 377–441.

[102] J.-R. Li, J. White, Low rank solution of lyapunov equations, SIAM Journal on Matrix Analysis and Applications 24 (1) (2002) 260–280.

[103] P. Benner, R.-C. Li, N. Truhar, On the adi method for sylvester equations, Journal of Computational and Applied Mathematics 233 (4) (2009) 1035–1045.

[104] M. Soltaniyeh, I. Kadayif, O. Ozturk, Classifying data blocks at subpage granularity with an on-chip page table to improve coherence in tiled cmps, IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems 37 (4) (2017) 806–819.

[105] S. Gao, I. W. Tsang, L. T. Chia, Kernel sparse representation for image classification and face recognition, in: Proceedings of the European Conference on Computer Vision (ECCV), Vol. 6314, 2010.

[106] H. V. Nguyen, V. M. Patel, N. M. Nasrabadi, R. Chellappa, Design of non-linear kernel dictionaries for object recognition, IEEE Transactions on Image Processing 22 (12) (2013) 5123–5135.

[107] H. Qi, S. Hughes, Using the kernel trick in compressive sensing: Accurate signal recovery from fewer measurements, in: IEEE International Conference on Acoustics, Speech and Signal Processing, 2011, pp. 3940 –3943.

[108] V. M. Patel, R. Vidal, Kernel sparse subspace clustering, in: IEEE International Conference on Image Processing, 2014.

[109] A. S. Georghiades, P. N. Belhumeur, D. J. Kriegman, From few to many: Illumination cone models for face recognition under variable lighting and pose, IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI) 23 (6) (2001) 643–660.

[110] A. M. Martinez, R. Benavente, AR face database, Tech. Rep. 24, CVC technical report (June 1998).

[111] S. Shekhar, V. M. Patel, N. M. Nasrabadi, R. Chellappa, Joint sparse representation for robust multimodal biometrics recognition, IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI) 36 (1) (2014) 113–126.

[112] M. E. Fathy, V. M. Patel, R. Chellappa, Face-based active authentication on mobile devices, in: IEEE International Conference on Acoustics, Speech and Signal Processing, 2015, pp. 1687–1691.

[113] H. Wang, S. Z. Li, Y. Wang, Face recognition under varying lighting conditions using self quotient image, in: IEEE International Conference on Automatic Face and Gesture Recognition, 2004, pp. 819–824.

[114] A. Krizhevsky, I. Sutskever, G. E. Hinton, Imagenet classification with deep convolutional neural networks, in: Advances in Neural Information Processing Systems (NeurIPS), 2012, pp. 1097–1105.

[115] D. Kang, H. Han, A. K. Jain, S.-W. Lee, Nighttime face recognition at large standoff: Cross-distance and cross-spectral matching, Pattern Recognition 47 (12) (2014) 3750–3766.

[116] C.-G. Li, R. Vidal, et al., Structured sparse subspace clustering: A unified optimization framework., in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2015, pp. 277–286.

[117] C. You, D. Robinson, R. Vidal, Scalable sparse subspace clustering by orthogonal matching pursuit, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2016, pp. 3918–3927.

[118] C.-Y. Lu, H. Min, Z.-Q. Zhao, L. Zhu, D.-S. Huang, S. Yan, Robust and efficient subspace segmentation via least squares regression, in: Proceedings of the European Conference on Computer Vision (ECCV), Springer, 2012, pp. 347–360.

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

[120] J. Ngiam, A. Khosla, M. Kim, J. Nam, H. Lee, A. Y. Ng, Multimodal deep learning, in: International Conference on Machine Learning (ICML), 2011, pp. 689–696.

[121] N. Srivastava, R. R. Salakhutdinov, Multimodal learning with deep boltzmann machines, in: Advances in Neural Information Processing Systems (NeurIPS), 2012, pp. 2222–2230.

[122] D. Ramachandram, G. W. Taylor, Deep multimodal learning: A survey on recent advances and trends, IEEE Signal Processing Magazine 34 (6) (2017) 96–108.

[123] A. Mobiny, P. A. Cicalese, S. Zare, P. Yuan, M. Abavisani, C. C. Wu, J. Ahuja, P. M. de Groot, H. Van Nguyen, Radiologist-level covid-19 detection using ct scans with detail-oriented capsule networks, arXiv preprint arXiv:2004.07407 (2020).

[124] K. Raeisi, M. Mohebbi, M. Khazaei, M. Seraji, A. Yoonessi, Phase-synchrony evaluation of eeg signals for multiple sclerosis diagnosis based on bivariate empirical mode decomposition during a visual task, Computers in Biology and Medicine 117 (2020) 103596.

[125] A. Naghizadeh, S. Berenjian, B. Razeghi, S. Shahanggar, N. R. Pour, Preserving receiver's anonymity for circular structured p2p networks, in: 2015 12th Annual IEEE Consumer Communications and Networking Conference (CCNC), IEEE, 2015, pp. 71–76.

[126] A. Naghizadeh, S. Berenjian, E. Meamari, R. E. Atani, Structural-based tunneling: preserving mutual anonymity for circular p2p networks, International Journal of Communication Systems 29 (3) (2016) 602–619.

[127] A. Naghizadeh, Improving fairness in peer-to-peer networks by separating the role of seeders in network infrastructures, Turkish Journal of Electrical Engineering & Computer Sciences 24 (4) (2016) 2255–2266.

[128] 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.

[129] D. P. Kingma, J. Ba, Adam: A method for stochastic optimization, arXiv preprint arXiv:1412.6980 (2014).

[130] N. X. Vinh, J. Epps, J. Bailey, Information theoretic measures for clusterings comparison: Variants, properties, normalization and correction for chance, Journal of Machine Learning Research 11 (Oct) (2010) 2837–2854.

[131] W. M. Rand, Objective criteria for the evaluation of clustering methods, Journal of the American Statistical association 66 (336) (1971) 846–850.

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

[133] A. Naghizadeh, D. N. Metaxas, Meaningful distance for multivariate clustering, in: 2018 International Conference on Computational Science and Computational Intelligence (CSCI), IEEE, 2018, pp. 1149–1154.

[134] M. Abavisani, V. M. Patel, Multimodal sparse and low-rank subspace clustering, Information Fusion 39 (2018) 168 – 177. doi:http://dx.doi.org/10.1016/j.inffus.2017.05.002.
URL http://www.sciencedirect.com/science/article/pii/S1566253517303123

[135] V. M. Patel, H. V. Nguyen, R. Vidal, Latent space sparse subspace clustering, in: IEEE International Conference on Computer Vision (ICCV), 2013.

[136] M. Abavisani, V. M. Patel, Domain adaptive subspace clustering, in: Britain Machine Vision Conference, 2017.

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

[138] P. Y. Simard, D. Steinkraus, J. C. Platt, et al., Best practices for convolutional neural networks applied to visual document analysis., in: Icdar, Vol. 3, 2003.

[139] D. C. Cireşan, U. Meier, L. M. Gambardella, J. Schmidhuber, Deep, big, simple neural nets for handwritten digit recognition, Neural computation 22 (12) (2010) 3207–3220.

[140] E. D. Cubuk, B. Zoph, D. Mane, V. Vasudevan, Q. V. Le, Autoaugment: Learning augmentation policies from data, arXiv preprint arXiv:1805.09501 (2018).

[141] A. Naghizadeh, M. Abavisani, D. N. Metaxas, Greedy autoaugment, Pattern Recognition Letters 138 (2020) 624–630.

[142] E. D. Cubuk, B. Zoph, J. Shlens, Q. V. Le, Randaugment: Practical automated data augmentation with a reduced search space, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops, 2020, pp. 702–703.

[143] D. Ho, E. Liang, X. Chen, I. Stoica, P. Abbeel, Population based augmentation: Efficient learning of augmentation policy schedules, in: International Conference on Machine Learning (ICML), PMLR, 2019, pp. 2731–2741.

[144] X. Guo, E. Zhu, X. Liu, J. Yin, Deep embedded clustering with data augmentation, in: Asian Conference on Machine Learning, 2018, pp. 550–565.

[145] X. Guo, X. Liu, E. Zhu, X. Zhu, M. Li, X. Xu, J. Yin, Adaptive self-paced deep clustering with data augmentation, IEEE Transactions on Knowledge and Data Engineering (2019).

[146] P. Bachman, O. Alsharif, D. Precup, Learning with pseudo-ensembles, in: Advances in Neural Information Processing Systems (NeurIPS), 2014, pp. 3365–3373.

[147] M. Sajjadi, M. Javanmardi, T. Tasdizen, Regularization with stochastic transformations and perturbations for deep semi-supervised learning, in: Advances in Neural Information Processing Systems (NeurIPS), 2016, pp. 1163–1171.

[148] S. Laine, T. Aila, Temporal ensembling for semi-supervised learning, arXiv preprint arXiv:1610.02242 (2016).

[149] T. Simon, H. Joo, I. Matthews, Y. Sheikh, Hand keypoint detection in single images using multiview bootstrapping, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2017, pp. 1145–1153.

[150] I. Radosavovic, P. Dollár, R. Girshick, G. Gkioxari, K. He, Data distillation: Towards omni-supervised learning, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2018, pp. 4119–4128.

[151] Y. Tian, D. Krishnan, P. Isola, Contrastive multiview coding, arXiv preprint arXiv:1906.05849 (2019).

[152] P. J. Rousseeuw, Silhouettes: a graphical aid to the interpretation and validation of cluster analysis, Journal of computational and applied mathematics 20 (1987) 53–65.

[153] D. L. Davies, D. W. Bouldin, A cluster separation measure, IEEE transactions on pattern analysis and machine intelligence (2) (1979) 224–227.

[154] A. Naghizadeh, D. N. Metaxas, Condensed silhouette: An optimized filtering process for cluster selection in k-means, Procedia Computer Science 176 (2020) 205–214.

[155] R. Vidal, P. Favaro, Low rank subspace clustering (LRSC), Pattern Recognition Letters (2013).

[156] Y. Le, X. Yang, Tiny imagenet visual recognition challenge, 2015.

[157] T. DeVries, G. W. Taylor, Improved regularization of convolutional neural networks with cutout, arXiv preprint arXiv:1708.04552 (2017).

[158] Z. Zhong, L. Zheng, G. Kang, S. Li, Y. Yang, Random erasing data augmentation, arXiv preprint arXiv:1708.04896 (2017).

[159] J. Wright, A. Yang, A. Ganesh, S. Sastry, Y. Ma, Robust face recognition via sparse representation, IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI) 31 (2) (2009) 210 –227.

[160] M. Aharon, M. Elad, A. Bruckstein, et al., K-svd: An algorithm for designing over-complete dictionaries for sparse representation, IEEE Transactions on Signal Processing 54 (11) (2006) 4311.

[161] X. Mei, H. Ling, Robust visual tracking and vehicle classification via sparse representation, IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI) 33 (11) (2011) 2259–2272.

[162] S. Shekhar, V. M. Patel, N. M. Nasrabadi, R. Chellappa, Joint sparse representation for robust multimodal biometrics recognition, IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI) 36 (1) (2014) 113–126.

[163] M. Abavisani, M. Joneidi, S. Rezaeifar, S. B. Shokouhi, A robust sparse representation based face recognition system for smartphones, in: 2015 IEEE Signal Processing in Medicine and Biology Symposium (SPMB), IEEE, 2015, pp. 1–6.

[164] M. Joneidi, A. Zaeemzadeh, S. Rezaeifar, M. Abavisani, N. Rahnavard, Lfm signal detection and estimation based on sparse representation, in: 2015 49th Annual Conference on Information Sciences and Systems (CISS), IEEE, 2015, pp. 1–5.

[165] A. Shrivastava, V. M. Patel, R. Chellappa, Multiple kernel learning for sparse representation-based classification, IEEE Transactions on Image Processing 23 (7) (2014) 3013–3024.

[166] H. V. Nguyen, V. M. Patel, N. M. Nasrabadi, R. Chellappa, Design of non-linear kernel dictionaries for object recognition, IEEE Transactions on Image Processing 22 (12) (2013) 5123–5135.

[167] V. M. Patel, R. Vidal, Kernel sparse subspace clustering, in: 2014 IEEE International Conference on Image Processing (ICIP), 2014, pp. 2849–2853.

[168] V. M. Patel, H. V. Nguyen, R. Vidal, Latent space sparse subspace clustering, in: IEEE International Conference on Computer Vision (ICCV), 2013, pp. 225–232.

[169] L. Zhang, W. D. Zhou, P. C. Chang, J. Liu, Z. Yan, T. Wang, F. Z. Li, Kernel sparse representation-based classifier, IEEE Transactions on Signal Processing 60 (4) (2012) 1684 –1695.

[170] X. T. Yuan, S. Yan, Visual classification with multi-task joint sparse representation, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2010.

[171] H. Van Nguyen, V. M. Patel, N. M. Nasrabadi, R. Chellappa, Kernel dictionary learning, in: IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), 2012, IEEE, 2012, pp. 2021–2024.

[172] J. Yin, Z. Liu, Z. Jin, W. Yang, Kernel sparse representation based classification, Neurocomputing 77 (1) (2012) 120–128.

[173] A. Gammerman, V. Vovk, V. Vapnik, Learning by transduction, in: Proceedings of the Fourteenth conference on Uncertainty in artificial intelligence, Morgan Kaufmann Publishers Inc., 1998, pp. 148–155.

[174] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, L. Fei-Fei, Imagenet: A large-scale hierarchical image database, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Ieee, 2009, pp. 248–255.

[175] K. Simonyan, A. Zisserman, Very deep convolutional networks for large-scale image recognition, arXiv preprint arXiv:1409.1556 (2014).

[176] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, Z. Wojna, Rethinking the inception architecture for computer vision, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2016, pp. 2818–2826.

[177] K. He, X. Zhang, S. Ren, J. Sun, Deep residual learning for image recognition, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2016, pp. 770–778.

[178] G. Huang, Z. Liu, L. Van Der Maaten, K. Q. Weinberger, Densely connected convolutional networks, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2017, pp. 4700–4708.

[179] M. Abavisani, V. Patel, Domain adaptive subspace clustering., in: British Machine Vision Conference, BMVA, 2016.

[180] L. Zhang, M. Yang, X. Feng, Sparse representation or collaborative representation: Which helps face recognition?, in: IEEE International Conference on Computer Vision (ICCV), IEEE, 2011, pp. 471–478.

[181] M. Abavisani, M. Joneidi, S. Rezaeifar, S. B. Shokouhi, A robust sparse representation based face recognition system for smartphones, in: IEEE Signal Processing in Medicine and Biology Symposium (SPMB), 2015, pp. 1–6.

[182] M. Soltaniyeh, R. P. Martin, S. Nagarakatte, Synergistic cpu-fpga acceleration of sparse linear algebra, arXiv preprint arXiv:2004.13907 (2020).

[183] S. Shafiee, F. Kamangar, V. Athitsos, J. Huang, L. Ghandehari, Multimodal sparse representation classification with fisher discriminative sample reduction, in: 2014 IEEE International Conference on Image Processing (ICIP), 2014, pp. 5192–5196. doi:10.1109/ICIP.2014.7026051.

[184] G. Goswami, P. Mittal, A. Majumdar, M. Vatsa, R. Singh, Group sparse representation based classification for multi-feature multimodal biometrics, Information Fusion 32 (2016) 3–12.

[185] H. Zhang, N. M. Nasrabadi, Y. Zhang, T. S. Huang, Multi-observation visual recognition via joint dynamic sparse representation, in: IEEE International Conference on Computer Vision (ICCV), IEEE, 2011, pp. 595–602.

[186] X.-T. Yuan, X. Liu, S. Yan, Visual classification with multitask joint sparse representation, IEEE Transactions on Image Processing 21 (10) (2012) 4349–4360.

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

[188] X. Wang, D. Kumar, N. Thome, M. Cord, F. Precioso, Recipe recognition with large multimodal food dataset, in: 2015 IEEE International Conference on Multimedia & Expo Workshops (ICMEW), IEEE, 2015, pp. 1–6.

[189] J. Devlin, M.-W. Chang, K. Lee, K. Toutanova, Bert: Pre-training of deep bidirectional transformers for language understanding, in: Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers), 2019, pp. 4171–4186.

[190] S. S. Rautaray, A. Agrawal, Vision based hand gesture recognition for human computer interaction: a survey, Artificial Intelligence Review 43 (1) (2015) 1–54.

[191] N. C. Camgoz, S. Hadfield, O. Koller, R. Bowden, Subunets: End-to-end hand shape and continuous sign language recognition, in: IEEE International Conference on Computer Vision (ICCV), IEEE, 2017, pp. 3075–3084.

[192] Z. Lv, A. Halawani, S. Feng, S. Ur Réhman, H. Li, Touch-less interactive augmented reality game on vision-based wearable device, Personal and Ubiquitous Computing 19 (3-4) (2015) 551–567.

[193] N. Neverova, C. Wolf, G. W. Taylor, F. Nebout, Multi-scale deep learning for gesture detection and localization, in: Workshop at the European conference on computer vision, Springer, 2014, pp. 474–490.

[194] P. Molchanov, S. Gupta, K. Kim, J. Kautz, Hand gesture recognition with 3d convolutional neural networks, in: Proceedings of the IEEE conference on computer vision and pattern recognition workshops, 2015, pp. 1–7.

[195] C. Li, X. Zhang, L. Jin, Lpsnet: A novel log path signature feature based hand gesture recognition framework, in: Computer Vision Workshop (ICCVW), 2017 IEEE International Conference on, IEEE, 2017, pp. 631–639.

[196] L. Zhang, G. Zhu, P. Shen, J. Song, S. A. Shah, M. Bennamoun, Learning spatiotemporal features using 3dcnn and convolutional lstm for gesture recognition, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2017, pp. 3120–3128.

[197] H. Wang, P. Wang, Z. Song, W. Li, Large-scale multimodal gesture recognition using heterogeneous networks, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2017, pp. 3129–3137.

[198] Q. Miao, Y. Li, W. Ouyang, Z. Ma, X. Xu, W. Shi, X. Cao, Z. Liu, X. Chai, Z. Liu, et al., Multimodal gesture recognition based on the resc3d network., in: ICCV Workshops, 2017, pp. 3047–3055.

[199] P. Molchanov, S. Gupta, K. Kim, K. Pulli, Multi-sensor system for driver's hand-gesture recognition, in: Automatic Face and Gesture Recognition (FG), 2015 11th IEEE International Conference and Workshops on, Vol. 1, IEEE, 2015, pp. 1–8.

[200] J. Wang, Z. Liu, J. Chorowski, Z. Chen, Y. Wu, Robust 3d action recognition with random occupancy patterns, in: Proceedings of the European Conference on Computer Vision (ECCV), Springer, 2012, pp. 872–885.

[201] X. Shen, G. Hua, L. Williams, Y. Wu, Dynamic hand gesture recognition: An exemplar-based approach from motion divergence fields, Image and Vision Computing 30 (3) (2012) 227–235.

[202] A. Tamrakar, S. Ali, Q. Yu, J. Liu, O. Javed, A. Divakaran, H. Cheng, H. Sawhney, Evaluation of low-level features and their combinations for complex event detection in open source videos, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), IEEE, 2012, pp. 3681–3688.

[203] D. Tran, L. Bourdev, R. Fergus, L. Torresani, M. Paluri, Learning spatiotemporal features with 3d convolutional networks, in: IEEE International Conference on Computer Vision (ICCV), 2015, pp. 4489–4497.

[204] K. Simonyan, A. Zisserman, Two-stream convolutional networks for action recognition in videos, in: Advances in Neural Information Processing Systems (NeurIPS), 2014, pp. 568–576.

[205] N. C. Camgoz, S. Hadfield, O. Koller, R. Bowden, Using convolutional 3d neural networks for user-independent continuous gesture recognition, in: Pattern Recognition (ICPR), 2016 23rd International Conference on, IEEE, 2016, pp. 49–54.

[206] R. Cui, H. Liu, C. Zhang, Recurrent convolutional neural networks for continuous sign language recognition by staged optimization, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2017.

[207] G. Zhu, L. Zhang, P. Shen, J. Song, Multimodal gesture recognition using 3-d convolution and convolutional lstm, IEEE Access 5 (2017) 4517–4524.

[208] S. J. Pan, Q. Yang, et al., A survey on transfer learning, IEEE Transactions on knowledge and data engineering 22 (10) (2010) 1345–1359.

[209] L. Torrey, J. Shavlik, Transfer learning, in: Handbook of Research on Machine Learning Applications and Trends: Algorithms, Methods, and Techniques, IGI Global, 2010, pp. 242–264.

[210] Y. Bengio, Deep learning of representations for unsupervised and transfer learning, in: Proceedings of ICML Workshop on Unsupervised and Transfer Learning, 2012, pp. 17–36.

[211] M. Oquab, L. Bottou, I. Laptev, J. Sivic, Learning and transferring mid-level image representations using convolutional neural networks, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2014, pp. 1717–1724.

[212] A. Karpathy, G. Toderici, S. Shetty, T. Leung, R. Sukthankar, L. Fei-Fei, Large-scale video classification with convolutional neural networks, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2014, pp. 1725–1732.

[213] J.-T. Huang, J. Li, D. Yu, L. Deng, Y. Gong, Cross-language knowledge transfer using multilingual deep neural network with shared hidden layers, in: Acoustics, Speech and Signal Processing (ICASSP), 2013 IEEE International Conference on, IEEE, 2013, pp. 7304–7308.

[214] J. Yosinski, J. Clune, Y. Bengio, H. Lipson, How transferable are features in deep neural networks?, in: Advances in Neural Information Processing Systems (NeurIPS), 2014, pp. 3320–3328.

[215] P. Perera, V. Patel, Deep transfer learning for multiple class novelty detection, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2019.

[216] R. Caruana, Multitask learning, Machine learning 28 (1) (1997) 41–75.

[217] P. Oza, V. M. Patel, Deep cnn-based multi-task learning for open-set recognition, arXiv preprint arXiv:1903.03161 (2019).

[218] P. Perera, M. Abavisani, V. M. Patel, In2i: Unsupervised multi-image-to-image translation using generative adversarial networks, in: 2018 24th International Conference on Pattern Recognition (ICPR), IEEE, 2018, pp. 140–146.

[219] L. A. Gatys, A. S. Ecker, M. Bethge, Image style transfer using convolutional neural networks, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), IEEE, 2016, pp. 2414–2423.

[220] B. Sun, K. Saenko, Deep coral: Correlation alignment for deep domain adaptation, in: Proceedings of the European Conference on Computer Vision (ECCV), Springer, 2016, pp. 443–450.

[221] B. Sun, J. Feng, K. Saenko, Correlation alignment for unsupervised domain adaptation, in: Domain Adaptation in Computer Vision Applications, Springer, 2017, pp. 153–171.

[222] L. Gatys, A. S. Ecker, M. Bethge, Texture synthesis using convolutional neural networks, in: Advances in Neural Information Processing Systems (NeurIPS), 2015, pp. 262–270.

[223] G. Farnebäck, Two-frame motion estimation based on polynomial expansion, in: Scandinavian conference on Image analysis, Springer, 2003, pp. 363–370.

[224] S. Ioffe, C. Szegedy, Batch normalization: Accelerating deep network training by reducing internal covariate shift, in: International Conference on Machine Learning (ICML), 2015, pp. 448–456.

[225] W. Kay, J. Carreira, K. Simonyan, B. Zhang, C. Hillier, S. Vijayanarasimhan, F. Viola, T. Green, T. Back, P. Natsev, et al., The kinetics human action video dataset, arXiv preprint arXiv:1705.06950 (2017).

[226] H. Wang, D. Oneata, J. Verbeek, C. Schmid, A robust and efficient video representation for action recognition, International Journal of Computer Vision 119 (3) (2016) 219–238.

[227] D. Tran, H. Wang, L. Torresani, J. Ray, Y. LeCun, M. Paluri, A closer look at spatiotemporal convolutions for action recognition, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2018, pp. 6450–6459.

[228] A. G. Howard, M. Zhu, B. Chen, D. Kalenichenko, W. Wang, T. Weyand, M. Andreetto, H. Adam, Mobilenets: Efficient convolutional neural networks for mobile vision applications, arXiv preprint arXiv:1704.04861 (2017).

[229] K. He, G. Gkioxari, P. Dollár, R. Girshick, Mask r-cnn, in: IEEE International Conference on Computer Vision (ICCV), 2017, pp. 2961–2969.

[230] J. Devlin, M.-W. Chang, K. Lee, K. Toutanova, Bert: Pre-training of deep bidirectional transformers for language understanding, arXiv preprint arXiv:1810.04805 (2018).

[231] C. Harding, F. Pompei, D. Burmistrov, H. G. Welch, R. Abebe, R. Wilson, Breast cancer screening, incidence, and mortality across us counties, JAMA internal medicine 175 (9) (2015) 1483–1489.

[232] T. Gebru, J. Krause, Y. Wang, D. Chen, J. Deng, E. L. Aiden, L. Fei-Fei, Using deep learning and google street view to estimate the demographic makeup of neighborhoods across the united states, Proceedings of the National Academy of Sciences 114 (50) (2017) 13108–13113.

[233] R. Abebe, S. Hill, J. W. Vaughan, P. M. Small, H. A. Schwartz, Using search queries to understand health information needs in africa, in: Proceedings of the International AAAI Conference on Web and Social Media, Vol. 13, 2019, pp. 3–14.

[234] S. Buechel, A. Buffone, B. Slaff, L. Ungar, J. Sedoc, Modeling empathy and distress in reaction to news stories, arXiv preprint arXiv:1808.10399 (2018).

[235] J. C. Eichstaedt, R. J. Smith, R. M. Merchant, L. H. Ungar, P. Crutchley, D. Preoţiuc-Pietro, D. A. Asch, H. A. Schwartz, Facebook language predicts depression in medical records, Proceedings of the National Academy of Sciences 115 (44) (2018) 11203–11208.

[236] S. C. Guntuku, D. Preotiuc-Pietro, J. C. Eichstaedt, L. H. Ungar, What twitter profile and posted images reveal about depression and anxiety, arXiv preprint arXiv:1904.02670 (2019).

[237] J. C. Eichstaedt, H. A. Schwartz, S. Giorgi, M. L. Kern, G. Park, M. Sap, D. R. Labarthe, E. E. Larson, M. Seligman, L. H. Ungar, et al., More evidence that twitter language predicts heart disease: A response and replication (2018).

[238] N. Said, K. Ahmad, M. Regular, K. Pogorelov, L. Hassan, N. Ahmad, N. Conci, Natural disasters detection in social media and satellite imagery: a survey, arXiv preprint arXiv:1901.04277 (2019).

[239] S. Madichetty, M. Sridevi, Detecting informative tweets during disaster using deep neural networks, in: 2019 11th International Conference on Communication Systems & Networks (COMSNETS), IEEE, 2019, pp. 709–713.

[240] T. G. Rudner, M. Rußwurm, J. Fil, R. Pelich, B. Bischke, V. Kopacková, Rapid computer vision-aided disaster response via fusion of multiresolution, multisensor, and multitemporal satellite imagery.

[241] T. Blevins, R. Kwiatkowski, J. MacBeth, K. McKeown, D. Patton, O. Rambow, Automatically processing tweets from gang-involved youth: Towards detecting loss and aggression, in: Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers, The COLING 2016 Organizing Committee, Osaka, Japan, 2016, pp. 2196–2206.
URL https://www.aclweb.org/anthology/C16-1207

[242] S. Kumar, G. Barbier, M. A. Abbasi, H. Liu, Tweettracker: An analysis tool for humanitarian and disaster relief, in: Fifth international AAAI conference on weblogs and social media, 2011.

[243] H. Shekhar, S. Setty, Disaster analysis through tweets, in: 2015 International Conference on Advances in Computing, Communications and Informatics (ICACCI), IEEE, 2015, pp. 1719–1723.

[244] K. Stowe, M. J. Paul, M. Palmer, L. Palen, K. Anderson, Identifying and categorizing disaster-related tweets, in: Proceedings of The Fourth International Workshop on Natural Language Processing for Social Media, 2016, pp. 1–6.

[245] H. To, S. Agrawal, S. H. Kim, C. Shahabi, On identifying disaster-related tweets: Matching-based or learning-based?, in: 2017 IEEE Third International Conference on Multimedia Big Data (BigMM), IEEE, 2017, pp. 330–337.

[246] J. Yin, S. Karimi, A. Lampert, M. Cameron, B. Robinson, R. Power, Using social media to enhance emergency situation awareness, in: Twenty-fourth international joint conference on artificial intelligence, 2015.

[247] K. Ahmad, M. Riegler, K. Pogorelov, N. Conci, P. Halvorsen, F. De Natale, Jord: a system for collecting information and monitoring natural disasters by linking social media with satellite imagery, in: Proceedings of the 15th International Workshop on Content-Based Multimedia Indexing, ACM, 2017, p. 12.

[248] X. Li, D. Caragea, H. Zhang, M. Imran, Localizing and quantifying damage in social media images, in: 2018 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining (ASONAM), IEEE, 2018, pp. 194–201.

[249] G. Nalluru, R. Pandey, H. Purohit, Relevancy classification of multimodal social media streams for emergency services, in: 2019 IEEE International Conference on Smart Computing (SMARTCOMP), IEEE, 2019, pp. 121–125.

[250] M. Wöllmer, A. Metallinou, F. Eyben, B. Schuller, S. Narayanan, Context-sensitive multimodal emotion recognition from speech and facial expression using bidirectional lstm modeling, in: Proc. INTERSPEECH 2010, Makuhari, Japan, 2010, pp. 2362–2365.

[251] S. Chen, Q. Jin, Multi-modal dimensional emotion recognition using recurrent neural networks, in: Proceedings of the 5th International Workshop on Audio/Visual Emotion Challenge, ACM, 2015, pp. 49–56.

[252] A. Naghizadeh, S. Berenjian, D. J. Margolis, D. N. Metaxas, Gnm: Gridcell navigational model, Expert Systems with Applications 148 (2020) 113217.

[253] R. Bernardi, R. Cakici, D. Elliott, A. Erdem, E. Erdem, N. Ikizler-Cinbis, F. Keller, A. Muscat, B. Plank, Automatic description generation from images: A survey of models, datasets, and evaluation measures, Journal of Artificial Intelligence Research 55 (2016) 409–442.

[254] T. Rahman, B. Xu, L. Sigal, Watch, listen and tell: Multi-modal weakly supervised dense event captioning, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2019, pp. 8908–8917.

[255] A. Fukui, D. H. Park, D. Yang, A. Rohrbach, T. Darrell, M. Rohrbach, Multimodal compact bilinear pooling for visual question answering and visual grounding, arXiv preprint arXiv:1606.01847 (2016).

[256] R. Socher, M. Ganjoo, C. D. Manning, A. Ng, Zero-shot learning through cross-modal transfer, in: Advances in Neural Information Processing Systems (NeurIPS), 2013, pp. 935–943.

[257] A. Frome, G. S. Corrado, J. Shlens, S. Bengio, J. Dean, M. Ranzato, T. Mikolov, Devise: A deep visual-semantic embedding model, in: Advances in Neural Information Processing Systems (NeurIPS), 2013, pp. 2121–2129.

[258] K.-H. Lee, X. Chen, G. Hua, H. Hu, X. He, Stacked cross attention for image-text matching, in: Proceedings of the European Conference on Computer Vision (ECCV), 2018, pp. 201–216.

[259] J. Lu, D. Batra, D. Parikh, S. Lee, Vilbert: Pretraining task-agnostic visiolinguistic representations for vision-and-language tasks, arXiv preprint arXiv:1908.02265 (2019).

[260] L. H. Li, M. Yatskar, D. Yin, C.-J. Hsieh, K.-W. Chang, Visualbert: A simple and performant baseline for vision and language, arXiv preprint arXiv:1908.03557 (2019).

[261] H. Tan, M. Bansal, Lxmert: Learning cross-modality encoder representations from transformers, arXiv preprint arXiv:1908.07490 (2019).

[262] W. Su, X. Zhu, Y. Cao, B. Li, L. Lu, F. Wei, J. Dai, Vl-bert: Pre-training of generic visual-linguistic representations, arXiv preprint arXiv:1908.08530 (2019).

[263] Z.-Z. Lan, L. Bao, S.-I. Yu, W. Liu, A. G. Hauptmann, Multimedia classification and event detection using double fusion, Multimedia tools and applications 71 (1) (2014) 333–347.

[264] A. Ilyas, Microfilters: Harnessing twitter for disaster management, in: IEEE Global Humanitarian Technology Conference (GHTC 2014), IEEE, 2014, pp. 417–424.

[265] S. Kelly, X. Zhang, K. Ahmad, Mining multimodal information on social media for increased situational awareness (2017).

[266] H. Mouzannar, Y. Rizk, M. Awad, Damage identification in social media posts using multimodal deep learning., in: ISCRAM, 2018.

[267] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, I. Polosukhin, Attention is all you need, in: Advances in Neural Information Processing Systems (NeurIPS), 2017, pp. 5998–6008.

[268] J. Hessel, B. Pang, Z. Zhu, R. Soricut, A case study on combining asr and visual features for generating instructional video captions, arXiv preprint arXiv:1910.02930 (2019).

[269] K. Liu, Y. Li, N. Xu, P. Natarajan, Learn to combine modalities in multimodal deep learning, arXiv preprint arXiv:1805.11730 (2018).

[270] I. Ilievski, J. Feng, Multimodal learning and reasoning for visual question answering, in: I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, R. Garnett (Eds.), Advances in Neural Information Processing Systems (NeurIPS), Curran Associates, Inc., 2017, pp. 551–562.
URL http://papers.nips.cc/paper/6658-multimodal-learning-and-reasoning-fo
pdf

[271] L. Wu, S. Li, C.-J. Hsieh, J. L. Sharpnack, Stochastic shared embeddings: Data-driven regularization of embedding layers, in: Advances in Neural Information Processing Systems (NeurIPS), 2019, pp. 24–34.

[272] M. Tan, Q. V. Le, Efficientnet: Rethinking model scaling for convolutional neural networks, arXiv preprint arXiv:1905.11946 (2019).

[273] A. Radford, K. Narasimhan, T. Salimans, I. Sutskever, Improving language understanding by generative pre-training, URL https://s3-us-west-2. amazonaws. com/openai-assets/researchcovers/languageunsupervised/language understanding paper. pdf (2018).

[274] Z. Yang, Z. Dai, Y. Yang, J. Carbonell, R. Salakhutdinov, Q. V. Le, Xlnet: Generalized autoregressive pretraining for language understanding, arXiv preprint arXiv:1906.08237 (2019).

[275] Anonymous, {ALBERT}: A lite {bert} for self-supervised learning of language representations, in: Submitted to International Conference on Learning Representations, 2020, under review.
URL https://openreview.net/forum?id=H1eA7AEtvS

[276] W. Inc., Bert trained on bookcorpus and english wikipedia data, https://resources.wolframcloud.com/NeuralNetRepository/resources/BERT-Trained-on-BookCorpus-and-English-Wikipedia-Data, accessed: 2020-03-30.

[277] F. Alam, F. Ofli, M. Imran, Crisismmd: Multimodal twitter datasets from natural disasters, in: Twelfth International AAAI Conference on Web and Social Media, 2018.

[278] D. Kiela, S. Bhooshan, H. Firooz, D. Testuggine, Supervised multimodal bitransformers for classifying images and text, arXiv preprint arXiv:1909.02950 (2019).

[279] H. Zhang, M. Cisse, Y. N. Dauphin, D. Lopez-Paz, mixup: Beyond empirical risk minimization, in: International Conference on Learning Representations, 2018.
URL https://openreview.net/forum?id=r1Ddp1-Rb

[280] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, R. Salakhutdinov, Dropout: a simple way to prevent neural networks from overfitting, The journal of machine learning research 15 (1) (2014) 1929–1958.

[281] S. Castro, D. Hazarika, V. Pérez-Rosas, R. Zimmermann, R. Mihalcea, S. Poria, Towards multimodal sarcasm detection (an _Obviously_ perfect paper), in: Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics, Association for Computational Linguistics, Florence, Italy, 2019, pp. 4619–4629. doi:10.18653/v1/P19-1455.
URL https://www.aclweb.org/anthology/P19-1455

[282] R. Schifanella, P. de Juan, J. Tetreault, L. Cao, Detecting sarcasm in multimodal social platforms, Proceedings of the 2016 ACM on Multimedia Conference - MM '16 (2016). doi:10.1145/2964284.2964321.
URL http://dx.doi.org/10.1145/2964284.2964321

[283] V. M. Patel, R. Gopalan, R. Li, R. Chellappa, Visual domain adaptation: A survey of recent advances, IEEE Signal Processing Magazine 32 (3) (2015) 53–69.

[284] K. Saenko, B. Kulis, M. Fritz, T. Darrell, Adapting visual category models to new domains, in: Proceedings of the European Conference on Computer Vision (ECCV), Springer, 2010, pp. 213–226.

[285] D. Yoo, N. Kim, S. Park, A. S. Paek, I. S. Kweon, Pixel-level domain transfer, in: Proceedings of the European Conference on Computer Vision (ECCV), Springer, 2016, pp. 517–532.

[286] I.-H. Jhuo, D. Liu, D. Lee, S.-F. Chang, Robust visual domain adaptation with low-rank reconstruction, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), IEEE, 2012, pp. 2168–2175.

[287] S. Shekhar, V. M. Patel, H. V. Nguyen, R. Chellappa, Generalized domain-adaptive dictionaries, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2013, pp. 361–368.

[288] R. Gopalan, R. Li, R. Chellappa, Unsupervised adaptation across domain shifts by generating intermediate data representations, IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI) 36 (11) (2014) 2288–2302.

[289] K. Bousmalis, N. Silberman, D. Dohan, D. Erhan, D. Krishnan, Unsupervised pixel-level domain adaptation with generative adversarial networks, arXiv preprint arXiv:1612.05424 (2016).

[290] Y. Taigman, A. Polyak, L. Wolf, Unsupervised cross-domain image generation, arXiv preprint arXiv:1611.02200 (2016).

[291] E. Tzeng, J. Hoffman, K. Saenko, T. Darrell, Adversarial discriminative domain adaptation, arXiv preprint arXiv:1702.05464 (2017).

[292] M. Long, H. Zhu, J. Wang, M. I. Jordan, Unsupervised domain adaptation with residual transfer networks, in: Advances in Neural Information Processing Systems (NeurIPS), 2016, pp. 136–144.