UNIVERSITY OF TECHNOLOGY SYDNEY
Faculty of Engineering and Information Technology

Common and Unique Feature Learning for Data Fusion

by

Sunny Verma

A Thesis Submitted
in Partial Fulfillment of the
Requirements for the Degree

Doctor of Philosophy

Sydney, Australia

2020

Certificate of Authorship/Originality

I, Sunny Verma declare that this thesis, is submitted in fulfilment of the requirements for award of Doctoral of Philosophy, in the School of Software, Faculty of Engineering and Information Technology at the University of Technology Sydney.

This thesis is wholly my own work unless otherwise reference or acknowledged. In addition, I certify that all information sources and literature used are indicated in this thesis.

This document has not been submitted for qualification at any other academic institution.

This research is supported by the Australian Government Research Training Program.

Signature: Production Note:
Signature removed prior to publication.

Date: 02/03/2020

Sunny Verma

Acknowledgements

First and foremost, I want to thank my supervisors Dr. Wei Liu and Dr. Chen Wang, for their support throughout my degree. The journey as a Ph.D. student is not for everyone. Only you know about the feeling of joy after acceptance of your work and the feeling of lost after a rejection. I could not have achieved half of what I have without strong supervision of my supervisors.

I would like to thank Prof. Liming Zhu and Data61, CSIRO for providing me the scholarship for this degree. I'm also thankful to my former supervisor Dr. Meng Hui Lim for the opportunity under his guidance in Hong Kong, and to many things beyond words.

I also appreciate the support and discussion with my friends, in particular, Dr. Abhinav Anand, Dr. Rajan Kashyap, Dr. Avinash Singh, and Yurui Ming. In a similar spirit, my colleagues at UTS and Data61, CSIRO have been a constant source of valuable moments I can remember.

Finally, I owe my family the irreversible time and patience they have had during all these years and thank Sau Kei Kwok for all her support.

Sunny Verma

Sydney, Australia, 2020.

List of Publications

Journal Papers

J-1. **Sunny Verma**, Chen Wang, Liming Zhu, and Wei Liu, "Attn-HybridNet: Enhancing Discriminablity of Features with Attention Fusion," *IEEE Transactions on Cybernetics*, **Under Review**.

Conference Papers

C-1. **Sunny Verma**, Chen Wang, Liming Zhu, and Wei Liu, "DeepCU: Integrating Both Common and Unique Latent Information for Multimodal Sentiment Analysis, *IJCAI19*, Aug 2019. **Accepted.**

C-2. **Sunny Verma**, Chen Wang, Liming Zhu, and Wei Liu, "Towards Effective Data Augmentation via Unbiased GAN Utilization, *PRICAI19*, Aug 2019. **Accepted.**

C-3. **Quan Do**, **Sunny Verma**, Fang Chen, and Wei Liu, "Multiple Knowledge Transfer for Cross Domain Recommendation, *PRICAI19*, Aug 2019. **Accepted.**

C-4. **Sunny Verma**, Chen Wang, Liming Zhu, and Wei Liu, "HybridNet: Improving Deep Learning Networks via Integrating Two Views of Images, *ICONIP18*, Dec 2018. **Accepted. Accepted.**

Conference Posters

P-1. **Sunny Verma**, Chen Wang, Liming Zhu, and Wei Liu, "A Compliance Checking Framework for DNN Models, *IJCAI19*, Aug 2019. **Accepted.**

P-2. **Sunny Verma**, Chen Wang, Liming Zhu, and Wei Liu, " Extracting Highly Effective Features for Supervised Learning via Simultaneous Tensor Factorization, *AAAI17*, Feb 2017. **Accepted.**

Contents

List of Figures

List of Tables

ABSTRACT

Common and Unique Feature Learning for Data Fusion

In today's era of big data, information about a phenomenon of interest is available from multiple acquisitions. Data captured from each of these acquisition frameworks are commonly known as modality, where each modality provides information in a complementary manner. Despite the evident benefits and plethora of works on data fusion, two challenging issues persist, 1) feature representation: how to exploit the data diversity that multiple modalities offer, and 2) feature fusion: how to combine the heterogeneous information for better decision making.

To address these challenges, this thesis presents a significantly improved model of two widely utilised fusion techniques, a) early fusion: combining features from multiple modalities for joint prediction, and b) late fusion: combining modality-specific predictions at the decision level. I illustrate how both these techniques have their own specific limitations, with late fusion unable to harness the inter-modality benefits, and the reliance of early fusion on a single model causing failure when information from any modality is futile. To overcome these drawbacks, I developed novel multimodal systems that performs feature extraction and feature fusion in a consolidated frameworks. Technically, I designed feature extraction schemes to capture both unique information from individual modalities and common information from multimode representations. I then combine these two kinds of information for supervised prediction, by designing efficient fusion schemes that enable this frameworks to perform information discovery and feature fusion simultaneously.

In this thesis, I also demonstrated the benefits of fusing both the common and unique information in supervised learning and validate the significance of the developed techniques on multimodal, multiview, and multisource datasets. The designed methods leverage the multimodal benefits by creating additional diversity, and ob-

tain a more unified view of the underlying phenomenon for better decision making.

Chapter 1

Introduction

The performance of any classification system is inherently affected by the feature representation utilised to build them. As different feature representations unveil a range of explanatory factors concealed in the data, considerable research effort has been dedicated to identify and obtain these factors. For this reason, feature engineering has remained an important research direction in the field of data mining and machine learning, and its dominance is evident in prominent conferences such as *ICDM*, *WWW*, and *SIGKDD*, etc. Furthermore, learning representations from data remain a critical task, information obtained from a single source is insufficient to represent the complexities of the underlying phenomenon [6, 5]. Signals from various acquisition frameworks, called modalities, promise a better understanding of the phenomenon of interest.

Despite being complementary, the availability of information from multiple modalities comes with two challenging issues, 1) why do we need their fusion, and 2) how to perform the fusion? The advantages of performing fusion are recursively proved in several domains for example, in speaker identification novel methods such as the works of Ren et. al. [97], have advanced the field by developing a multimodal system to encapsulate time dependencies from both the visual and auditory modalities. However, how to efficiently perform this fusion is an active research topic, and its challenges and opportunities are amplified with the proliferation of devices generating data, and with the advancement in artificial intelligence (AI) techniques utilising them.

In this thesis, I present research on common and unique feature learning for data fusion. In particular, I first demonstrate that the two kinds of information that is common information that is available from joint analysis of modalities and unique information, which is available from independently analysing a single modality, are both essential, but individually insufficient, for supervised classification.

I then designed novel algorithms to obtain both the common and the unique information and illustrate how these two kinds of information are complementary while performing fusion.

1.1 Outline

The first component of this research is presented in Chapter 3, where I demonstrate how utilising both common and unique information can address the current bottleneck of obtaining lightweight deep neural networks. Despite achieving great success in image recognition, state of the art deep neural networks require high-end hardware, and this restricts their general-purpose utilisation. While multiple solutions have been proposed in the literature [39, 16], they all compress an already trained deep neural network, rather than develop a lightweight network. At first I designed a **HybridNet** that extracts the unique information from each view* and common information by combining all the views of an image. I then demonstrate how these two kinds of information are visually different but are complementary to each other. Finally, these two kinds of information are combined, in the proposed attention-based fusion scheme in **Attn-HybridNet**. Multiple performance metrics are then empirically evaluated, to discuss how our proposed framework is independent of high-end hardware and to validate the superiority of utilising both common and unique information over either common or unique information.

*Mode of an image, for example, a *RGB* image has three views: height, width, and color channels.

In Chapter 4, we propose a deep neural network that utilises both common and unique information to perform multimodal sentiment analysis. Multimodal sentiment analysis combines information available from visual, textual, and acoustic representations for sentiment prediction [129]. Recenty developed multimodal fusion schemes utilise an outer product on the individual modalities, to obtain their multimodal representation as a tensor. These schemes either obtain common information from the multimodal representation, by training a feed-forward neural network on the tensor, or they obtain unique information by modeling low-rank representation for each mode of the tensor independently. In this research, for multimodal data we show that both the common information and the unique information are essential, as they render inter-modal and intra-modal relationships of the data. This insight derived proposal of a) a novel deep architecture as a common network to extract the common information from the multi-mode representations, and b) unique networks to obtain the modality-specific information that enhances the generalisation performance of this multimodal system. Finally, both common and unique information is integrated via a fusion layer, in a novel multimodal data fusion architecture, called as **DeepCU** (Deep network with both common and unique latent information). The proposed **DeepCU** consolidates the two networks for joint utilisation and discovery of all-important information, and is shown to be beneficial over current state of the art approaches for multimodal sentiment analysis.

In Chapter 5, I address the problem of dataset bias and its adverse effects on machine learning algorithms. Effects of various known dataset biases, such as data imbalance, and presence of sensitive attribute can be handled by following standard mitigation protocols [82]. However, many types of subtle bias (especially in the case of unstructured datasets, such as images) exist in datasets, for example selection bias, capture bias, and the label bias [112]. Such biases remain undetected in the datasets and their effects on the predictor are catastrophic. The goal of this research

is to reduce the bias learned by the machine learning algorithms, by formulating it as a data augmentation problem [88]. Any unknown (or undetected) bias present in the dataset is reflected as the common distribution by the predictor's learning mechanism† [2]. We augment the datasets with synthetic data instances, by devising policies to increase the starved unique distribution of within-class examples. The devised data provisioning mechanism promises that synthetic examples selected for data augmentation reduces the bias and the variance in learning mechanisms of supervised predictors.

1.2 Contributions of this Thesis

The main contributions of this thesis are as follows.

1. The development of a lightweight deep network called **HybridNet** which simultaneously extracts the unique information from the amalgamated view and the common information from the minutiae view of the images. I first propose a deep network based on tensor factorisation, called the **Tensor Factorisation Networks**, to extract the common information and design the custom-built *Left One Mode Out Orthogonal Iteration* (*LoMOI*) method to obtain weights of its convolution-tensor filters. The unique information is obtained utilising the **PCANet**, that uses the *principal components* to obtain weights of its convolution matrix filters. I then demonstrate that these two kinds of information are essential, but individually insufficient for classification. The proposed **HybridNet** integrates the information discovery and feature extraction from both views of the data in its consolidated architecture, which is independent of high-performance hardware for image classification.

†As the latent representation of a dataset reflects its underlying distribution.

2. The problem of feature redundancy in the **HybridNet** is addressed by designing an attention-based fusion scheme called the **Attn-HybridNet**. The **HybridNet** utilises generalised spatial pooling operation to aggregate the feature maps from its convolution layers which incur redundancy in the feature representations and are unable to accommodate the spatial structure of the natural images. To eradicate this, the proposed **Attn-HybridNet** performs feature selection and aggregation with an attention fusion, and enhances the discriminability of the hybrid features. The main advantages of **Attn-HybridNet** are that it decouples the feature extraction, and feature aggregation processes and requires significantly less computational resources for training.

3. A proposal of a novel deep neural network for multimodal data fusion called **DeepCU**. Initially it will be shown that how the common and the unique information in multimodal datasets are complementary, as they render inter-modal and intra-modal relationships of the data. I then propose two novel deep networks to extract information from a) the multi-mode representations (the common network) and b) information from individual modalities (unique networks). Finally, I integrate these two aspects of information via a fusion layer in the novel multimodal data fusion deep network, **DeepCU**.

4. The problem of dataset bias in deep neural networks is alleviated by proposing a data provisioning mechanism named as **Data Augmentation Pursuit** (**DAP**). The provisioning mechanism in **DAP** is composed of two sequential stages, Stage-1: labelled synthetic image generation with GANs [95], and Stage-2: iterative image filtering to sieve unbiased synthetic examples. The retained synthetic data instances obtained after Stage-2 of the **DAP** are utilized to augment the training datasets. Deep neural networks trained on augmented datasets obtained using **DAP** achieve significantly better classification per-

formance and exhibit a reduction in the bias and the variance in its learning mechanism.

1.3 Thesis Organisation

The remainder of this thesis is organised as follows. In Chapter 2, a review the of the existing fusion techniques, background on tensor decomposition and synthetic data generation using generative adversarial networks (GANs) is provided. Chapter 3 presents the contribution of extracting common and unique information from images and their attention based fusion. In Chapter 4, a brief literature review is provided, and details of our deep common and unique latent information fusion for multimodal sentiment analysis. In Chapter 5, the consequences of dataset bias are discussed and the necessity of developing the proposed data provisioning mechanism is framed. Chapter 6 presents the conclusions of this research and provides directions for future work.

Chapter 2

Background

In this chapter, we briefly introduce widely utilized multimodal fusion schemes and provide literature review on tensors and generative adversarial networks (GAN). The tensor decomposition framework is utilized in the Chapter. 3 and Chapter. 4, whereas the GANs are utilized in Chapter. 5.

2.1 Multimodal Data Fusion

Multimodal approaches are key elements in various disciplines [114, 103, 108] as the world around us is composed of multiple modalities such as we utilize our visual and auditory sensors during communication. In general, the term modality is used to indicate the data (signals) associated with an event, or phenomenon of interest. Therefore, a research problem is characterised as multimodal when it includes learning from multiple modalities [5]. Such as we can predict the sentiment of a speaker by utilizing the visual i.e., facial expressions, vocal intonations, and the transcript of the spoken utterances.

Similarly, when the research problem involves utilisation of different types of information from the same modality, it is referred to as multiview [70]li2018survey. Such as we can utilize both the temporal and contextual information from the transcript of the spoken utterances to perform sentiment prediction. Furthermore, when the research involves data from multiple sources such as web images and local * images than it is characterized as multisource learning [30].

*Images from personal devices or anything else than web.

Learning from heterogeneous data is vital as complementary information is available within different aspects, and in principle, capable of more robust inferences [67, 5]. The literature on multimodal fusion either belong to early or late fusion, typically depending on the task as there is no consensus on which fusion is the best as the level of fusion depends on the task at hand. However, a non-trivial task in all aspects of fusion is feature extraction. Since a good representation[†] is responsible for the success of data fusion [6]. Hence, a general conjecture can be made that feature extraction, and feature fusion goes hand in hand.

I will introduce the literature on feature extraction on the different data fusion tasks in the respective chapter. However, in this chapter, I will briefly summarise the two widely utilized data fusion techniques. Since tensors play a central role for feature extraction in this thesis, I also provide details of tensors decomposition and its applications in the subsequent sections.

2.1.1 Early Fusion

Early fusion can be seen as one of the earliest and widely utilized techniques to perform multimodal fusion. Techniques employing early fusion schemes create a joint representation of the input features (or models) from multiple modalities and train a single model for prediction. These schemes rely on a single model to learn the correlation between low-level features i.e., inter-modality interactions. Since the assumption is a single model is well suited for all the modalities. It requires the features from all the modalities to be highly engineered and aligned for efficient learning. The general scheme for early fusion is illustrated in Fig. 2.1

Early fusion schemes are seen as true multimodal learning schemes as the features from multiple modalities are combined from the beginning. Early fusion enjoys the

[†]A representation that is useful as input to the classifier

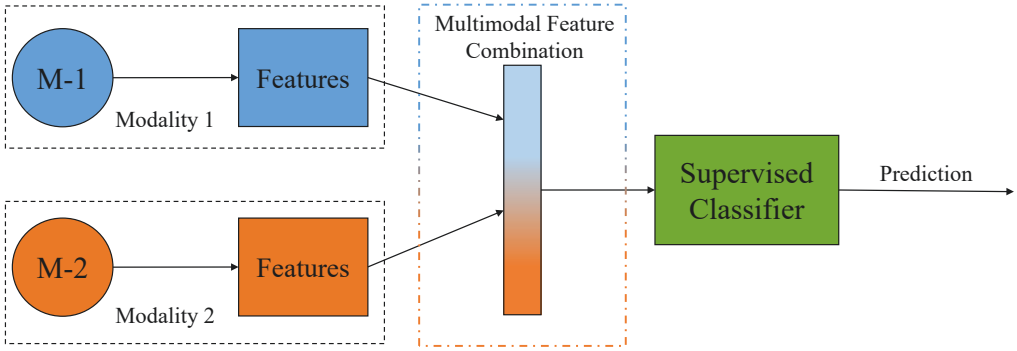


Figure 2.1 : Workflow of Early Fusion Scheme.

advantages of a single learning scheme. However, combining multiple heterogeneous representations is usually tricky, and is as a disadvantage of this scheme.

2.1.2 Late Fusion

Late fusion schemes utilize unimodal decision values and combine them with a fusion mechanism such as averaging, voting [84], or a learned model [96] for the final prediction. It allows the flexibility to utilize different models for each modality, thus allowing late fusion to learn semantic concepts from unimodal features. The individual models in late fusion rely on supervised learning to classify the semantic concepts from individual modalities. The general scheme for late fusion is illustrated in Fig. 2.2

Since predictions are made separately on individual modalities and hence it is easier to deal with scenarios when features from some modality is missing. However, late fusion is not as effective as early fusion at modeling inter-modality correlations as the fusion is performed at the decision level.

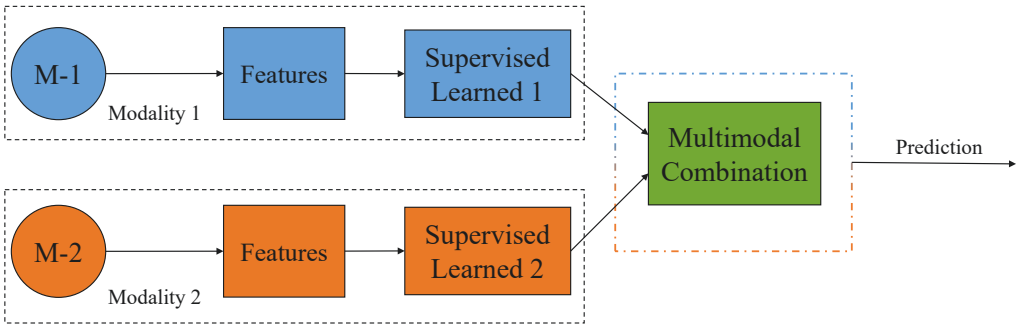


Figure 2.2 : Workflow of Late Fusion Scheme.

2.2 Tensor

Tensors are higher-order generalizations of matrices and are denoted by boldface Euler letters such as $\mathcal{X}, \mathcal{Y}, \mathcal{Z}$ etc. The number of modes also called order of tensor is equal to the number of dimensions of tensor $\mathcal{X}$. Formally a tensor $\mathcal{X}$ is usually of order 3 or greater where the first-order tensor $\boldsymbol{x}$ is the vector, and a second-order tensor $\boldsymbol{X}$ is a matrix. Many kinds of data frequently encountered in machine learning/data mining naturally occur in the form of tensors, for example, an *RGB* image is a third-order tensor with height, width, and depth as its modes. While tensors were originally introduced in psychometrics by Hitchcock in 1927 [45] but their utilization is expanded to multiple domains, like data science, machine learning, and statistics [20]. Fueled by low computational complexity and effectiveness in discovering dependencies in multi-dimensional data tensors and their applications have tremendously increased in deep learning. We now introduce a few important concepts with tensors in the next subsection.

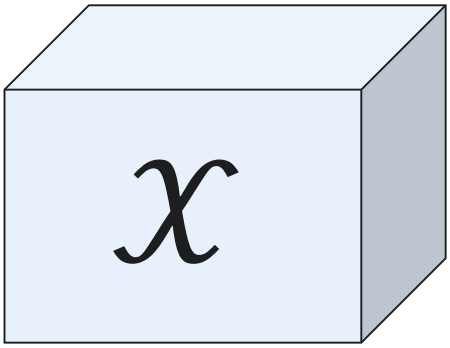


Figure 2.3 : A third order tensor, $\mathcal{X}$.

2.2.1 Tensor Preliminaries

Tensor Operations

We begin our introduction with tensors with elementary operations related to general purpose utilization of tensors and then gradually move to the introduction of existing tensor factorization techniques. However for ease of understanding, a third order tensor $\mathcal{X}$ is shown in Fig. 2.3

Tensor Indexing. Individual elements in a third-order tensor $\mathcal{X} \in \mathbb{R}^{I \times J \times K}$ are denotes as $\boldsymbol{x}_{ijk}$. It is analogous to matrix and vector indexing scheme where the individual elements are denoted as $\boldsymbol{X}_{ij}$ and $\boldsymbol{x}_i$, respectively. One can also extract subarrays from a tensor by fixing a subset of indices of the tensor. For a third-order tensor, we have the concept of fibers and slices defined in the next paragraphs.

Tensor Fibers. Fibers are higher-order analog to rows and columns of a matrix and are created by indexing tensor on a single mode. For a third-order tensor $\mathcal{X}$ we can extract column, row, and tube fibers are denoted as $\boldsymbol{x}_{:jk}$, $\boldsymbol{x}_{i:k}$, and $\boldsymbol{x}_{ij:}$ respectively, where the colon indicates all elements of a mode. Fig. 2.4 illustrate fibers of a third order tensor $\mathcal{X}$.

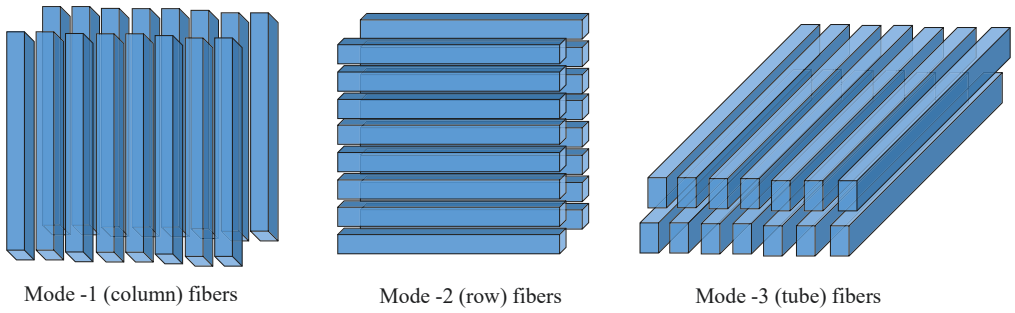


Figure 2.4 : Fibers of third order tensor $\mathcal{X}$.

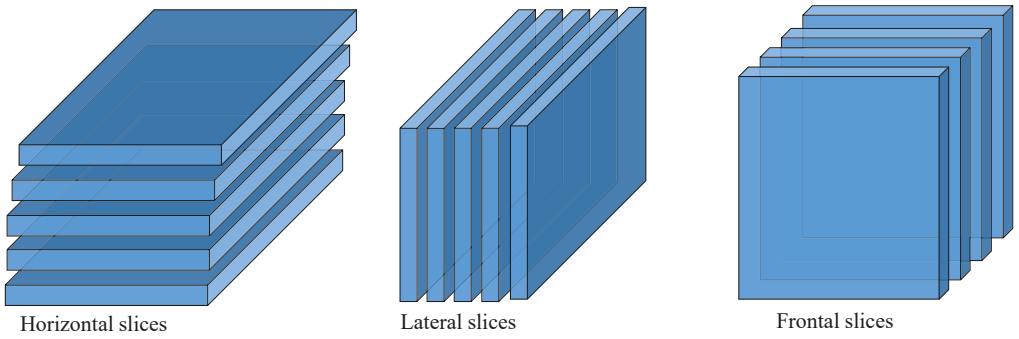


Figure 2.5 : Slices of third order tensor $\mathcal{X}$.

Tensor Slices. For an N-order tensor, a tensor slice represents (N-1)-order section of the tensor. For example, for a third-order tensor, $\mathcal{X}$ a slice will represent a two-dimensional section. For a third-order tensor $\mathcal{X}$ we have the horizontal, lateral, and frontal slices denoted as $\boldsymbol{X}_{i::}$, $\boldsymbol{X}_{:j:}$, and $\boldsymbol{X}_{::k}$ respectively.

Matriciziation. Also known as tensor unfolding, is the operation to rearrange the elements of an n-mode tensor $\mathcal{X} \in \mathbb{R}^{i_1 \times i_2 \ldots \times i_N}$ as matrix $\mathbf{X}_{(n)} \in \mathbb{R}^{i_n \times j}$ on the chosen mode n, where $j = \big(i_1 \ \ldots \times i_{n-1} \times i_{n+1} \ldots \times i_N\big)$. Fig. 2.5 illustrate slices of a third order tensor $\mathcal{X}$.

n-mode Product. The product of an n-mode tensor $\mathcal{X} \in \mathbb{R}^{i_1 \ldots \times i_{m-1} \times i_m \times i_{m+1} \ldots \times i_n}$ and a matrix $\mathbf{A} \in \mathbb{R}^{j \times i_n}$ is denoted as $\mathcal{X} \times_n \mathbf{A}$. The resultant of this product is also a tensor $\mathcal{Y} \in \mathbb{R}^{i_1 \times i_2 \times i_{n-1} \times j \times i_{n+1} \ldots \times i_n}$ which can also be expressed through matricized

tensor as $\mathbf{Y}_{(n)} = \mathbf{A}\mathbf{X}_{(n)}$.

Tensor Inner Product. The inner product between two tensors of same-order is the sum of the products of their entries. For example inner product between two third-order tensors $\mathcal{X}$ and $\mathcal{Y}$ is defined as $\left\langle \mathcal{X}, \mathcal{Y} \right\rangle = \sum_{i=1}^{I} \sum_{j=1}^{J} \sum_{k=1}^{K} \boldsymbol{x}_{ijk} \boldsymbol{y}_{ijk}$.

Tensor Norm. The tensor norm is analog to matrix and vector norm and is defined as the square root of the sum of the squares of all its elements. For a third-order tensor $\mathcal{X}$ its norm is defined as $\parallel \mathcal{X} \parallel = \sqrt{\sum_{i=1}^{I} \sum_{j=1}^{J} \sum_{k=1}^{K} \boldsymbol{x}_{ijk} \boldsymbol{y}_{ijk}}$; it can also be defined as $\parallel \mathcal{X} \parallel = \sqrt{\left\langle \mathcal{X}, \mathcal{X} \right\rangle}$.

With the preliminaries defined as above, we now move to our next subsection on tensor decomposition and their applications.

2.2.2 Tensor Decomposition Algorithms

Tensor decomposition is a form of generalized matrix factorization for approximating multimode tensors. While a plethora of tensor decomposition algorithms are available in the literature however all of them can be categorized in these two decomposition families: 1) canonical polyadic decomposition popularly known as the CP decomposition [45, 56] and 2) Tucker decomposition also known as $HOSVD$ [113, 27]. The CP decomposition expresses a tensor as the sum of a finite number of rank-one tensors [62] and is usually advised for estimation of latent factor, whereas the Tucker decomposition is a form of higher-order PCA [76] and is mostly applied for compression and dimensionality reduction. We only provide details of Tucker decomposition in this chapter as our contributions in this thesis are aligned with tensor compression and hence reviewing CP decomposition is futile. However, an interested reader can refer to the seminal work of Kolda and Barder [62].

Algorithm 1 Higher Order Orthogonal Iteration, *HOOI*

Input: n-mode tensor $\mathcal{X} \in \mathbb{R}^{i_1, i_2, ..., i_n}$; factorization ranks for each mode of the tensor $[r_1, r_2, ..., r_n]$, where $r_k \leq i_k \forall\ k \in 1, 2, ..., n$; factorization error-tolerance ε, and maximum allowable iterations $= N$

for $i = 1, 2, ..., n$ **do**

  initialize $\mathbf{U}^{(i)} \in \mathbb{R}^{i_i \times r_i}$ using $HOSVD$

$\mathcal{G} \leftarrow \mathcal{X} \times_1 (\mathbf{U}^{(1)})^T ... \times_m (\mathbf{U}^{(m)})^T ... \times_n (\mathbf{U}^{(n)})^T$ ▷ obtain core-tensor

$\hat{\mathcal{X}} \leftarrow \mathcal{G} \times_1 \mathbf{U}^{(1)} ... \times_m \mathbf{U}^{(m)} ... \times_n \mathbf{U}^{(n)}$ ▷ reconstructed tensor

$loss \leftarrow \|\mathcal{X} - \hat{\mathcal{X}}\|$ ▷ decomposition loss

$count \leftarrow 0$

while $\left[(loss \geq \varepsilon)\ Or\ (N \leq count)\right]$ **do** ▷ loop until convergence

  for $i = 1, 2, ..., n$ **do**

    $\mathcal{Y} \leftarrow \mathcal{X} \times_1 (\mathbf{U}^{(1)})^T ... \times_{(i-1)} (\mathbf{U}^{(i-1)})^T \times_{(i+1)} (\mathbf{U}^{(i+1)})^T ... \times_n (\mathbf{U}^{(n)})^T$ ▷ obtain the variance in mode-i

    $\mathbf{Y}_i \leftarrow$ unfold tensor $\mathcal{Y}$ on mode-i

    $\mathbf{U}^{(i)} \leftarrow \mathbf{r}_i$ left singular vectors of $\mathbf{Y}_i$

  $\mathcal{G} \leftarrow \mathcal{X} \times_1 (\mathbf{U}^{(1)})^T ... \times_m (\mathbf{U}^{(m)})^T ... \times_n (\mathbf{U}^{(n)})^T$

  $\hat{\mathcal{X}} \leftarrow \mathcal{G} \times_1 \mathbf{U}^{(1)} ... \times_m \mathbf{U}^{(m)} ... \times_n \mathbf{U}^{(n)}$

  $loss \leftarrow \|\mathcal{X} - \hat{\mathcal{X}}\|$

  $count \leftarrow count + 1$

Output: Factor matrices for each mode of the tensor i.e., $\left[\mathbf{U}^{(1)} ... \mathbf{U}^{(m)} ... \mathbf{U}^{(n)}\right]$

Tucker Decomposition

The Tucker decomposition first introduced by Tucker in [113] and is a form of higher-order PCA. It factorizes an order-n tensor $\mathcal{X} \in \mathbb{R}^{i_1 \times i_2 ... \times i_n}$ to obtain two sub components: 1) $\mathcal{G} \in \mathbb{R}^{r_1 \times r_2 ... \times r_n}$ which is a lower dimensional tensor called the *core-tensor* and, 2) $\mathbf{U}^{(j)} \in \mathbb{R}^{r_n \times i_n},\ \forall j = 1, ..., n$ which are matrix factors associated with each mode of the tensor. The entries in the *core-tensor* $\mathcal{G}$ signifies the interaction level between tensor element, whereas the factor matrices $\mathbf{U}^{(n)}$ are usually orthogonal and are analogue to *principal components* associated with the respective mode-n. Due to the work in [26] the Tucker factorization scheme is now called as the higer-order SVD ($HOSVD$) as the authors has shown that $HOSVD$ as a generalization of matrix SVD in their work.

In order to obtain the factor matrices $\mathbf{U}^{(i)}$ from tensor $\mathcal{X}$ we first unfold the tensor

on each mode and obtain the leading left singular vectors i.e. $\mathbf{U}^{(i)} \leftarrow SVD(\boldsymbol{X}_i)$, leading left singular vectors of $\boldsymbol{X}_i$. After obtaining matrix factors from each mode we can obtain the core tensor by performing multiplication of the tensor and matrix factors i.e., $\mathcal{G} = \boldsymbol{\mathcal{X}} \times_1 (\mathbf{U}^{(1)})^T ... \times_m (\mathbf{U}^{(m)})^T ... \times_n (\mathbf{U}^{(n)})^T$. The original tensor $\boldsymbol{\mathcal{X}}$ can be reconstructed by taking the n-mode product of the *core-tensor* and the factor matrices as in Eq. 2.1.

$$\mathcal{G} \times_1 \mathbf{U}^{(1)} \times_2 \mathbf{U}^{(2)} ... \times_N \mathbf{U}^{(n)} \approx \hat{\boldsymbol{\mathcal{X}}} \tag{2.1}$$

The *HOSVD* is computationally expensive in terms of obtaining singular vectors, and therefore its variant called the truncated *HOSVD* is more popular which obtain $r_n < i_n$ singular vectors from mode-n of the tensor. However, the truncated *HOSVD* is not optimal in terms of obtaining the best low rank approximation of the original tensor and *Higher Order Orthogonal Iteration* (*HOOI*) is proposed in [27]. The *HOOI* utilizes *HOSVD* as an initial solution and then iterates for computing the dominant subspace of the orthonormal basis. The pseudocode for obtaining matrix factors from an order-n tensor $\boldsymbol{\mathcal{X}}$ is presented in Alg. 1.

The advantages of utilizing *Tucker* based factorization methods have already been studied in several domains. In computer vision, [117] modeled the face recognition problem as multi-mode tensors and popularized them as Tensor-faces. In data mining, [102] formulated handwritten digits recognition through tensor factorization, whereas in signal processing, the works in [24, 20] considered the problem of brain signal analysis with tucker decomposition. Recent works in [63, 129, 47] etc. have proven the benefits of utilizing tensors and tucker decomposition in application of deep learning systems.

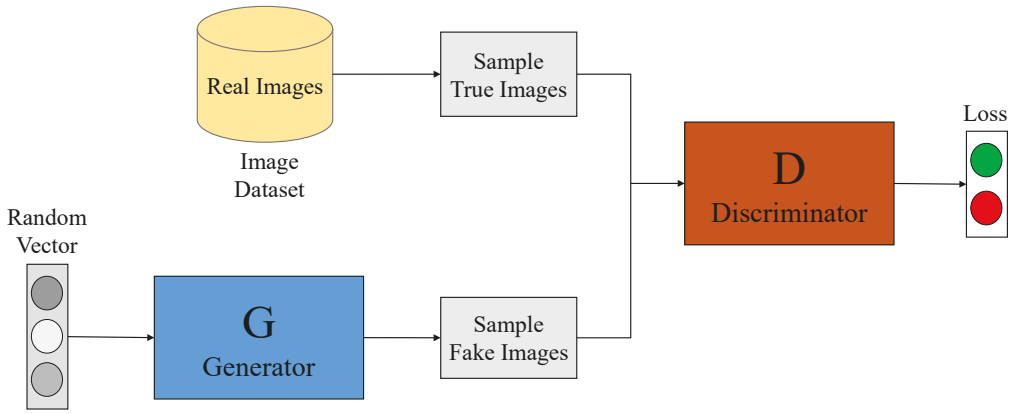


Figure 2.6 : Workflow of Generative Adversarial Networks.

2.3 Generative Adversarial Networks

Generative Adversarial Networks (**GAN**s) first introduced in [35] are typically composed of two deep neural networks as shown in Fig. 2.6. The first network is called the discriminator (**D**), while the second network is called the generator (**G**). Formally, the learning algorithm of GANs is a two-player zero-sum game where the loss (or gain) by the first player on the utility function is balanced by gain (or loss) of the utility by the second player [120]. The **GAN**s have become one of the most popular methods for generating synthetic images consisting of face, object, hand-written digits, etc. They are also extensively utilized in image to image translation [136], facial attribute manipulation [19, 79], text generation [125] and etc.

The generator network aims to generate photo-realistic images from randomly sampled noise prior z. In other words, if p_x is the distribution over true data then $\mathbf{G}(z)$ learns the distribution $p_g \sim p_x$. On the other hand, **D** aims at learning the discrimination between the distributions p_x and p_g, where $\mathbf{D}(\text{input})$ represents the probability $(p_x|input)$ and $\mathbf{G}(z)$ represents the output from **G** having noise (z) as its input. The optimization scheme for training the **D** and the **G** is performed via a

joint objective function $V(G, D)$ (minimax two-player game objective) as in Eq. 2.2.

$$\min_G \max_D V(G, D) = \mathbb{E}_{x \sim p_{data}(x)}\big[logD(x)\big] + \mathbb{E}_{z \sim p_{data}(z)}\big[log(1 - D(G(z)))\big] \quad (2.2)$$

The deep convolution GAN a.k.a *DCGAN* is believed to be the first **GAN** architecture which applied convolution to the generator and discriminator networks and generated high-quality images. The architecture of GANs has progressively evolved to solve a multitude of challenges faced while training **GAN**. Few notable works include *Wasserstein-GAN* [4, 37] and *BEGAN* [7] were proposed to solve the model collapse problems in **GAN**. While *LSGAN* [80] was proposed to address the non-convergence issue while training **GAN**s. A progressive strategy for generating high-resolution images with **GAN** is described in [54].

Moreover, few interesting applications of GANs include *CoGAN* (Coupled GAN) [72] which couple a pair of generative adversarial networks to learn a joint distribution over multiple modalities; this is achieved by sharing weights among higher convolution layers. Similarly, *InfoGAN* [15] is an information-theoretic extension allowing learning meaningful representations of objects with the GAN framework. While, in *CycleGAN* [135] allows style and domain transfer by learning cross-domain relationships. Furthermore works, like ImprovedGAN [100] extended the GAN framework for semi-supervised classification. Besides, currently, the generator is not capable of adding real-world flavors to the synthetic examples unless domain-specific operations as in [61, 105] are not applied to the synthetic images.

Despite the recent advancements in **GAN**, synthetic images generated by them on datasets with high variabilities like CIFAR [65] or ImageNet [29] are of low quality [37, 123]. Improving the quality of the images generated by the **GAN**s is currently an active research topic. A simple yet intuitive evaluation of synthetic images generated with **GAN**s is described in [104].

Chapter 3

Attn-HybridNet: Enhancing Discriminability of Hybrid Features with Attention Fusion

3.1 Introduction

Feature engineering is an essential task in the development of machine learning systems and has been well-studied with substantial efforts from communities including computer vision, data mining, and signal processing [134]. However, today, in the era of deep-learning, the features are extracted by processing the data through multiple stacked layers in deep neural networks. These deep neural networks sequentially perform sophisticated operations to obtain superior data representation by discovering critical information concealed in the data [6]. However, the training time required to obtain efficient representation from the data is exponentially large. Since the data-dependent optimization process of these networks is conditioned via stochastic optimization techniques which necessitates multiple flops of the data. Moreover, these deep networks have an exhaustive hyper-parameter search space during training and usually suffer from various training difficulties [33]. Besides, the deep networks are complex models and require high computational resources for their training and deployment. This limits their usability on micro-devices such as cellphones [87, 39]. The current research trend focuses on alleviating the memory and space requirements associated with the deep networks [63].

To produce lightweight convolution neural networks (CNNs) architecture, most of the existing solutions 1) approximate the convolution and fully connected layers by factorization[133, 69], or 2) compress the network layers with quantization/hashing

[39, 16] or, 3) replace the fully connected layer with a tensorized layer and optimize the weights of this layer by retraining [63]. However, these techniques require an already trained CNN network and hence can only work as post-optimization corrective procedures. Whereas, our objective is to build lightweight deep networks which are computationally inexpensive to train. In other words, in this research we seek the possibility of building deep networks which are independent of high-performance hardware and exhaustive hyperparameter search space required for their training.

3.2 Our Contributions

We summarize our contributions in this chapter as follows:

1. We propose **Tensor Factorized Network** (**TFNet**) which extracts features from the minutiae view of the data and is able to capture the spatial information present in the data that is beneficial for image classification.

2. We propose Left one Mode Out Orthogonal Iteration ($LoMOI$) algorithm for obtaining weights of convolution filters from the minutiae view of the data in **TFNet**.

3. We propose **Hybrid Network** (**HybridNet**) which integrates the feature extraction and information discovery procedure from two views of the data. The **HybridNet** reduces the information loss from the data by combing the merits of the **PCANet** and **TFNet** and obtains superior features than either of the two schemes.

4. We propose **Attn-HybridNet** for alleviating feature redundancy among the hybrid features by performing feature selection and aggregation with an attention-based fusion scheme. The **Attn-HybridNet** enhances the discriminability of the feature representations, which further boosts the classification performance.

5. We evaluate multiple case studies with the features obtained by **HybridNet** and **Attn-HybridNet** on CIFAR-10 dataset to demonstrate the effectiveness of our proposed fusion technique.

The rest of this chapter is organized as follows. We review the related literature consisting of **PCANet** and and background tensor preliminaries in Sec. 3.3. The details of our proposed **TFNet**, **HybridNet**, and **Attn-HybridNet** are presented in Sec. 3.4, Sec. 3.5, and Sec. 3.6 respectively. Sec. 3.7 describes the experimental setup and datasets utilized in this chapter followed by Sec. 3.8 for reporting the performances. We conclude this chapter with summaries in Sec. 3.9.

3.3 Literature Review

The success of utilizing CNNs for multiple computer vision tasks such as visual categorization, semantic segmentation, etc. has lead to drastic development in the field. However, to supersede the human performance on image classification tasks, these CNNs have grown tremendously deeper, for example, ResNet [42] achieving a top-5 error rate of 3.57% consisting of 152-layers, 60M parameters, and requiring 2.25×10^{10} flops at the time of inference. This restricts the applicability of these models on devices with limited computational resources such as mobile devices and reducing the size of the CNNs has become a non-trivial task for their practical applications.

Three main research directions are conducted in this regard 1) compression of CNNs with quantization, 2) approximating convolution layer with factorization, and 3) replacing fully connected layers with custom-built layers. The works in [39, 16, 14] accelerates and reduce the size of CNNs by compressing layers of the CNN models with quantization or hashing. These quantized CNN models achieve similar recognition accuracy with significantly less requirement for computational

resources during inference. Similarly, the works in [133, 69] obtain approximations of fully connected and convolution layers by utilizing factorization for compressing the CNN models. However, both the quantization and factorization based methods compresse a pre-trained CNN model instead of building a smaller or faster CNN model at the first place. Therefore, these techniques inherit the limitations of the pre-trained CNN models; for example, these compressed models might not adapt to images with different size and might require retraining for accommodating them.

However, the research works in [63, 87] take a slightly different approach than the above research directions and; propose new lightweight layers to replace fully connected layers which can efficiently reduce the size of any CNN model. The work in [63] propose a neural tensor layer while the work in [87] proposes a BoF (Bag-of-features) model as a neural pooling layer. These methods augment the CNNs and produce their lightweight versions which are trainable in an end-to-end fashion by backpropagation. However, a major limitation of these works is that they are only capable of replacing a fully connected layer and to replace a convolution layer. They work similarly to the approximation and quantization approaches.

A possible solution for obtaining lightweight CNNs architecture with lower computational requirements on smaller size images are proposed in **PCANet** [13] and **TFNN** [17]. The **PCANet** is a deep unsupervised parsimonious feature extractor whereas **TFNN** is a supervised CNN architecture utilizing neural tensor factorization for extracting information from multiway data. Both networks achieve very high classification performance on handwritten digits dataset but fail to obtain competitive performance on object recognition dataset. This is because the **PCANet** (and its later variant **FANet** [49]) incurs information loss associated with the spatial structure of the data as it obtains weights of its convolution filters from the amalgamated view of the data. Contrarily, the **TFNN** extracts information by isolating each view of the multi-view data and fails to efficiently consolidate them for their

utmost utilization incurring the loss of common information present in the data.

However, the information from both the amalgamated view and the minutiae view* are essential for classification, and their integration can enhance the classification performance [73, 118]. In this research, we first propose **HybridNet**, which integrates the two kinds of information in its deep parsimonious feature extraction architecture. A major difference between **HybridNet** and **PCANet** is that the former simultaneously obtains information from both views of the data whereas the latter is restricted to obtain information from the amalgamated view of the data. The **HybridNet** is also notably different from **TFNN** as the former is an unsupervised deep network while the latter is a supervised deep neural network. Moreover, the **HybridNet** extracts information from minutiae view of the data, whereas the **TFNN** extracts information by isolating each mode of multi-view data.

Later, to enhance the discriminability of the features obtained with **HybridNet**, we propose **Attn-HybridNet** which performs attention-based fusion on hybrid features. The **Attn-HybridNet** reduces the feature redundancy by performing feature selection and obtains superior feature representations for supervised classification. We present the related background on **PCANet** and tensor preliminaries in the next subsection. The differences and similarities between the **PCANet**, **TFNet**, **HybridNet**, and **Attn-HybridNet** are summarized in Table 3.1.

3.3.1 Background

We briefly summarize **PCANet**'s 2-layer architecture in this section.

*Throughout this thesis, we refer to the vectorized presentation of the data as the amalgamated view where all modes of the data (also called dimension for higher order-matrices i.e., tensors) are collapsed to obtain a vector. Whereas,the untransformed view of the data, i.e., when viewed with its multiple modes(e.g., tensors), is referred to as the minutiae view of the data.

Methods	Amalgamated View	Minutiae View	Attention Fusion
PCANet [13]	✓	×	×
TFNet [119]	×	✓	×
HybridNet [119]	✓	✓	×
Attn-HybridNet	✓	✓	✓

Table 3.1 : Comparison of different feature extraction models

The First Layer

The procedure begins by extracting overlapping patches of size $k_1 \times k_2$ around each pixel in the image; where patches from image I_i are denoted as $\mathbf{x}_{i,1}, \mathbf{x}_{i,2}, .., \mathbf{x}_{i,\tilde{m}\tilde{n}} \in \mathbb{R}^{k_1 k_2}$, where $\tilde{m} = m - \lceil \frac{k_1}{2} \rceil$[†] and $\tilde{n} = n - \lceil \frac{k_2}{2} \rceil$. Next, the obtained patches are zero-centered by subtracting the mean of the image patches and *vectorized* to obtain $\boldsymbol{X}_i \in \mathbb{R}^{k_1 k_2 \times \tilde{m}\tilde{n}}$ as the patch matrix. After repeating the same procedure for all the training images we obtain $\boldsymbol{X} \in \mathbb{R}^{k_1 k_2 \times N\tilde{m}\tilde{n}}$ as the final patch-matrix from which the *pca* filters are obtained. The *PCA* minimizes the reconstruction error with orthonormal filters known as the principal eigenvectors of $\boldsymbol{X}\boldsymbol{X}^T$ calculated as in Eq. 3.1

$$\min_{V \in \mathbb{R}^{k_1 k_2 \times L_1}} \|\boldsymbol{X} - VV^T\boldsymbol{X}\|_F, \ s.t. \quad V^T V = I_{L_1} \tag{3.1}$$

where I_{L_1} is an identity matrix of size $L_1 \times L_1$ and L_1 is the total number of obtained filters. These convolution filters can now be expressed as:

$$W^1_{l_{\mathbf{PCANet}}} = mat_{k_1,k_2}(ql(\boldsymbol{X}\boldsymbol{X}^T)) \in \mathbb{R}^{k_1 \times k_2} \tag{3.2}$$

where $mat_{k_1,k_2}(v)$ is a function that maps $v \in \mathbb{R}^{k_1 k_2}$ to a matrix $W \in \mathbb{R}^{k_1 \times k_2}$, and $ql(\boldsymbol{X}\boldsymbol{X}^T)$ denotes the l-th principal eigenvector of $\boldsymbol{X}\boldsymbol{X}^T$. Next, each training image

[†]The operator $\lceil z \rceil$ gives the smallest integer greater than or equal to z.

I_i is convolved with the L_1 filters as in Eq. 3.3.

$$I^l_{i_{\mathbf{PCANet}}} = I_i * W^1_{l_{\mathbf{PCANet}}} \tag{3.3}$$

where $*$ denotes the 2D convolution and i, l are the image and filter indices respectively. Importantly, the boundary of image I_i is padded before convolution to obtain $I^l_{i_{\mathbf{PCANet}}}$ with the same dimensions as in I_i. From Eq. 3.3 a total of $N \times L_1$ images are obtained and attributed as the output from the first layer.

The Second Layer

The methodology of the second layer is similar to the the first layer. We collect overlapping patches of size $k_1 \times k_2$ around each pixel from all input images in this layer i.e., from $I^l_{i_{\mathbf{PCANet}}}$. Next, we vectorize and zero-centre these images patches to obtain the final patch matrix denoted as $\boldsymbol{Y} \in \mathbb{R}^{k_1 k_2 \times L_1 N \tilde{m} \tilde{n}}$. This patch matrix is then utilized to obtain the convolution *pca* filters in layer 2 as in Eq. 3.4.

$$W^2_{l_{\mathbf{PCANet}}} = mat_{k_1,k_2}(ql(\boldsymbol{Y}\boldsymbol{Y}^T)) \in \mathbb{R}^{k_1 \times k_2} \tag{3.4}$$

where $l = [1, L_2]$ denotes the number of *pca* filters obtained in this layer. Next, the input images in this layer $I^l_{i_{\mathbf{PCANet}}}$ are convolved with the learned filters $W^2_{l_{\mathbf{PCANet}}}$ to obtain the output from this layer in Eq. 3.5. These images are then passed to the feature aggregation phase as in the next subsection.

$$O^l_{i_{\mathbf{PCANet}}} = I^l_{i_{\mathbf{PCANet}}} * W^2_{l_{\mathbf{PCANet}}} \tag{3.5}$$

The Output Layer

The output layer combines the output from all the convolution layers of **PCANet** to obtain the feature vectors. The process initiates by first binarizing each of the real-valued outputs from Eq. 3.5 by utilizing a Heaviside function $H(O^l_{i_{\mathbf{PCANet}}})$ on them, which converts the positive entries to 1 otherwise 0. Then, these L_2 outputs

are assembled into L_1 batches, where all images in a batch belong to the same convolution filter in the first layer. Then, these images are combined to form a single image by applying weighted sum as in Eq. 3.6 whose pixel value is in the range $[0,\ 2^{L_2}-1]$:

$$\mathcal{I}^{l}_{i_{\mathbf{PCANet}}} = \sum_{l=1}^{L_2} 2^{l-1} H(O^{2}_{l_{\mathbf{PCANet}}}) \tag{3.6}$$

Next, these binarized images are partitioned into B blocks and a histogram with 2^{L_2} bins is obtained. Finally, the histograms from all the B blocks are concatenated to form a feature vector from the amalgamated view of the images in Eq. 3.7.

$$f_{i_{\mathbf{PCANet}}} = [Bhist(\mathcal{I}^{1}_{i_{\mathbf{PCANet}}}), ..., Bhist(\mathcal{I}^{L_1}_{i_{\mathbf{PCANet}}})]^T \in \mathbb{R}^{(2^{L_2})L_1 B} \tag{3.7}$$

This block-wise encoding process encapsulates the L_1 images from Eq. 3.6 into a single feature vector which can be utilized for any machine learning task like clustering or classification.

3.3.2 Tensor Factorization using *LoMOI*

Tensors are simply multi-mode arrays or higher-order[‡] matrices of dimension > 2. In this chapter, the vectors are denoted as $\boldsymbol{x}$ are called first-order tensors, whereas the matrices are denoted as $\boldsymbol{X}$ are called second-order tensors. Analogously, matrices of order-3 or higher are called tensors and are denoted as $\boldsymbol{\mathcal{X}}$. Besides, multilinear algebraic operations such as *n-mode product* and *Matricization* are presented in Sec. 2.2.1.

To obtain weights of convolution-tensor filters we devise a custom-designed tucker-based tensor factorization scheme called as *Left one Mode Out Orthogonal Iteration* (*LoMOI*) presented in Alg. 2. The *LoMOI* obtains factors from each mode of the tensor but the sample mode. Obtaining matrix factors from the sample mode is futile

[‡]Also known as modes (dimensions) of a tensor and are analogous to rows and columns of a matrix.

Algorithm 2 Left One Mode Out Orthogonal Iteration, *LoMOI*

Input: n-mode tensor $\mathcal{X} \in \mathbb{R}^{i_1,i_2,\ldots,i_n}$; factorization ranks for each mode of the tensor $[r_1 \ldots r_{m-1}, r_{m+1} \ldots r_n]$, where $r_k \leq i_k \forall\ k \in 1,2,\ldots,n$ and $k \neq m$; factorization error-tolerance ε, and Maximum allowable iterations $= Maxiter$, $m =$ mode to discard while factorizing
for $i = 1,2,\ldots,n$ and $i \neq m$ **do**
  $\mathbf{X}_i \leftarrow$ unfold tensor $\mathcal{X}$ on mode-i
  $\mathbf{U}^{(i)} \leftarrow r_i$ left singular vectors of $\mathbf{X}_i$ ▷ extract leading r_i matrix factors
$\mathcal{G} \leftarrow \mathcal{X} \times_1 (\mathbf{U}^{(1)})^T \ldots \times_{m-1} (\mathbf{U}^{(m-1)})^T \times_{m+1} (\mathbf{U}^{(m+1)})^T \ldots \times_n (\mathbf{U}^{(n)})^T$ ▷ Core tensor
$\hat{\mathcal{X}} \leftarrow \mathcal{G} \times_1 \mathbf{U}^{(1)} \ldots \times_{m-1} \mathbf{U}^{(m-1)} \times_{m+1} \mathbf{U}^{(m+1)} \times_n \mathbf{U}^{(n)}$ ▷ reconstructed tensor obtained by multilinear product of the core-tensor with the factor-matrices.
$loss \leftarrow \|\mathcal{X} - \hat{\mathcal{X}}\|$ ▷ decomposition loss
$count \leftarrow 0$
while $[(loss \geq \varepsilon)\ Or\ (Maxiter \leq count)]$ **do** ▷ loop until convergence
  for $i = 1,2,\ldots,n$ and $i \neq m$ **do**
    $\mathcal{Y} \leftarrow \mathcal{X} \times_1 (\mathbf{U}^{(1)})^T \ldots \times_{(i-1)} (\mathbf{U}^{(i-1)})^T \times_{(i+1)} (\mathbf{U}^{(i+1)})^T \ldots \times_n (\mathbf{U}^{(n)})^T$ ▷ obtain the variance in mode-i
    $\mathbf{Y}_i \leftarrow$ unfold tensor $\mathcal{Y}$ on mode-i
    $\mathbf{U}^{(i)} \leftarrow \mathbf{r}_i$ left singular vectors of $\mathbf{Y}_i$
  $\mathcal{G} \leftarrow \mathcal{X} \times_1 (\mathbf{U}^{(1)})^T \ldots \times_{(m-1)} (\mathbf{U}^{(m-1)})^T \times_{(m+1)} (\mathbf{U}^{(m+1)})^T \ldots \times_n (\mathbf{U}^{(n)})^T$
  $\hat{\mathcal{X}} \leftarrow \mathcal{G} \times_1 \mathbf{U}^{(1)} \ldots \times_{(m-1)} \mathbf{U}^{(m-1)} \times_{(m+1)} \mathbf{U}^{(m+1)} \ldots \times_n \mathbf{U}^{(n)}$
  $loss \leftarrow \|\mathcal{X} - \hat{\mathcal{X}}\|$
  $count \leftarrow count + 1$
Output: $\hat{\mathcal{X}}$ the reconstructed tensor and $[\mathbf{U}^{(1)} \ldots \mathbf{U}^{(m-1)}, \mathbf{U}^{(m+1)} \ldots \mathbf{U}^{(n)}]$ the factor matrices

if we want to obtain dominant subspaces from the $RowView$ and the $ColumnView$ of an image. Hence, obtaining is not only futile but will also increase the computational complexity to the proposed **TFNet**.

3.4 The Tensor Factorization Network

The development of **Tensor Factorization Network** (**TFNet**) is motivated to reduce the loss of spatial information occurring in the **PCANet** while vectorizing image patches. However, this transformation of the data is inherent while extracting the *principal components* which destroys the geometric structure of the object encapsulated in the data which is proven beneficial in many image classification tasks [117, 118, 17]. Furthermore, the vectorization of the data results in high dimen-

sional vectors and generally requires more computational resources. Motivated by the above shortcomings with the **PCANet**, we propose the **TFNet**. The **TFNet** preserves the spatial structure of the data while obtaining weights of its convolution-tensor filters. The unsupervised feature extraction procedure from minutiae view of the data in **TFNet** is detailed in the next subsection.

3.4.1 The First Layer

Similar to the first layer in **PCANet**, we begin by collecting all overlapping patches of size $k_1 \times k_2$ around each pixel from the image I_i. However, contrary to **PCANet** the spatial structure of these patches are preserved and instead of matrix, we obtain a 3-mode tensor $\boldsymbol{\mathcal{X}}_i \in \mathbb{R}^{k_1 \times k_2 \times \tilde{m}\tilde{n}}$. The mode-1 and mode-2 of this tensor represents the row-space, and the column-space spanned by the pixels in the image. The mode-3 of this tensor represents the total number of image patches obtained from the input image. Iterating this process for all the training images, we obtain $\boldsymbol{\mathcal{X}} \in \mathbb{R}^{k_1 \times k_2 \times N\tilde{m}\tilde{n}}$ as our final patch-tensor. The matrix factors utilized to generate our convolution-tensorial filters for to the first two modes of $\boldsymbol{\mathcal{X}}$ are obtained by utilizing our custom-designed *LoMOI* (presented in Alg. 2) in Eq. 3.8.

$$[\hat{\boldsymbol{\mathcal{X}}}, \mathbf{U}^{(1)}, \mathbf{U}^{(2)}] \leftarrow LoMOI(\boldsymbol{\mathcal{X}}, r_1, r_2) \tag{3.8}$$

where $\hat{\boldsymbol{\mathcal{X}}} \in \mathbb{R}^{r_1 \times r_2 \times N\tilde{m}\tilde{n}}$, $\mathbf{U}^{(1)} \in \mathbb{R}^{k_1 \times r_1}$, and $\mathbf{U}^{(2)} \in \mathbb{R}^{k_2 \times r_2}$. We discard obtaining the matrix factors from mode-3 of tensor $\boldsymbol{\mathcal{X}}$ (which is $\boldsymbol{X}_3$) as this is equivalent to the transpose of the patches matrix $\boldsymbol{X}$ in layer 1 of the **PCANet** which is not factorized in the **PCANet** while obtaining weights for its convolution filters. Moreover, the matrix factors for this mode spans the sample space of the data which is trivial. A total of $L_1 = r_1 \times r_2$ convolution-tensor filters are obtained from the factor matrices $\mathbf{U}^{(1)}$ and $\mathbf{U}^{(2)}$ as in Eq. 3.9.

$$W^1_{l_{\mathbf{TFNet}}} = \mathbf{U}^{(1)}_{(:,i)} \otimes \mathbf{U}^{(2)}_{(:,j)} \in \mathbb{R}^{k_1 \times k_2} \tag{3.9}$$

where '$\otimes$' is the *outer*-product between two vectors, $i = [1, r_1]$, $j = [1, r_2]$, $l = [1, L_1]$, and $\mathbf{U}^{(m)}_{(:,i)}$ represents 'i^{th}' column of the 'm^{th}' factor matrix. Importantly, our convolution-tensorial filters do not require any explicit reshaping as the *outer*-product between two vectors naturally results in a matrix. Therefore, we can straightforwardly convolve the input images with our obtained convolution-tensorial filters as described in Eq. 3.10 where $i = [1, N]$ and $l = [1, L_1]$ are the image and filter indices respectively

$$I^{l}_{i_{\mathbf{TFNet}}} = I_i * W^{1}_{l_{\mathbf{TFNet}}} \tag{3.10}$$

However, whenever the data is an *RGB*-image, each extracted patch from the image is a 3-order tensor $\boldsymbol{\mathcal{X}} \in \mathbb{R}^{k_1 \times k_2 \times 3}$ (i.e., *RowPixels*$\times$*ColPixels*$\times$Color). After collecting patches from all the training images, we obtain a 4-mode tensor as $\boldsymbol{\mathcal{X}} \in \mathbb{R}^{k_1 \times k_2 \times 3 \times N\tilde{m}\tilde{n}}$ which is decomposed by utilizing *LoMOI* ($[\hat{\boldsymbol{\mathcal{X}}}, \mathbf{U}^{(1)}, \mathbf{U}^{(2)}, \mathbf{U}^{(3)}] \leftarrow LoMOI(\boldsymbol{\mathcal{X}}, r_1, r_2, r_3)$) for obtaining the convolution-tensorial filters in Eq. 3.11.

$$W^{1}_{l_{\mathbf{TFNet}}} = U^{(1)}_{(:,i)} \otimes U^{(2)}_{(:,j)} \otimes U^{(3)}_{(:,k)} \tag{3.11}$$

where $i \in [1, r_1]$, $j \in [1, r_2]$, and $k \in [1, r_3]$.

3.4.2 The Second Layer

Similar to the first layer, we extract overlapping patches from the input images and zero-center them to build a 3-mode patch-tensor denoted as $\boldsymbol{\mathcal{Y}} \in \mathbb{R}^{k_1 \times k_2 \times NL_1\tilde{m}\tilde{n}}$ which is decomposed as $[\hat{\boldsymbol{\mathcal{Y}}}, \mathbf{V}^{(1)}, \mathbf{V}^{(2)}] \leftarrow LoMOI(\boldsymbol{\mathcal{Y}}, r_1, r_2)$ to obtain the convolution-tensor filters for layer 2 in Eq. 3.12.

$$W^{2}_{l_{\mathbf{TFNet}}} = \mathbf{V}^{(1)}_{(:,i)} \otimes \mathbf{V}^{(2)}_{(:,j)} \in \mathbb{R}^{k_1 \times k_2} \tag{3.12}$$

where, $\hat{\boldsymbol{\mathcal{Y}}} \in \mathbb{R}^{r_1 \times r_2 \times NL_1\tilde{m}\tilde{n}}$, $\mathbf{V}^{(1)} \in \mathbb{R}^{k_1 \times r_1}$, and $\mathbf{V}^{(2)} \in \mathbb{R}^{k_2 \times r_2}$, $i = [1, r_1]$, $j = [1, r_2]$, and $l = [1, L_2]$. We, now convolve each of the L_1 input images from the first layer with the convolution-tensorial filters obtained as below in Eq. 3.13.

$$O^{l}_{i_{\mathbf{TFNet}}} = I^{l}_{i_{\mathbf{TFNet}}} * W^{2}_{l_{\mathbf{TFNet}}}, \quad l = 1, 2, ..., L_2 \tag{3.13}$$

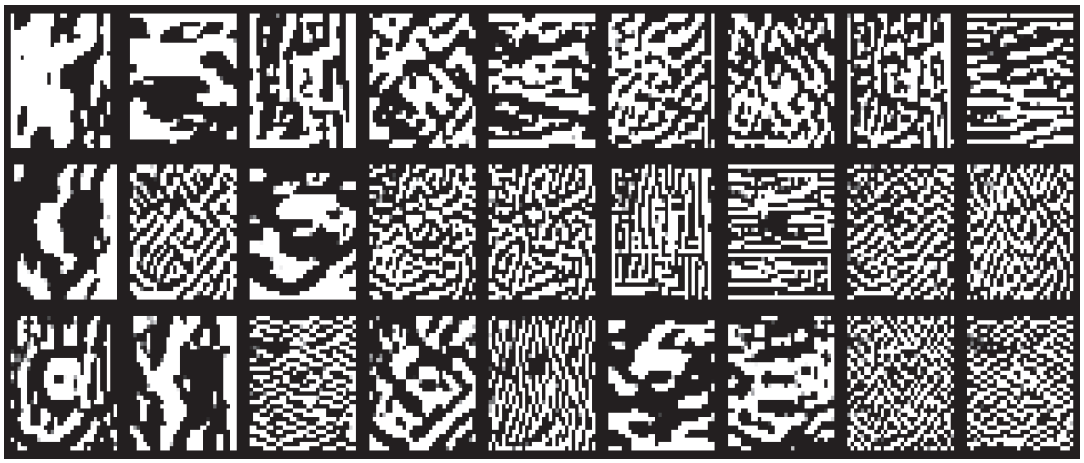
The number of output images obtained here is equal to $L_1 \times L_2$ which is identical to the number of images obtained at layer 2 of **PCANet**. Finally, we utilize the output layer of **PCANet** (Sec. 3.3.1) to obtain the feature vectors from the minutiae view of the image in Eq. 3.14.

$$\begin{aligned} \mathcal{I}^l_{i_{\mathbf{TFNet}}} &= \sum_{l=1}^{L_2} 2^{l-1} H(O^2_{l_{\mathbf{TFNet}}}) \\ f_{i_{\mathbf{TFNet}}} &= [Bhist(\mathcal{I}^1_{i_{\mathbf{TFNet}}}), ..., Bhist(\mathcal{I}^{L_1}_{i_{\mathbf{TFNet}}})]^T \in \mathbb{R}^{(2^{L_2})L_1 B} \end{aligned} \tag{3.14}$$

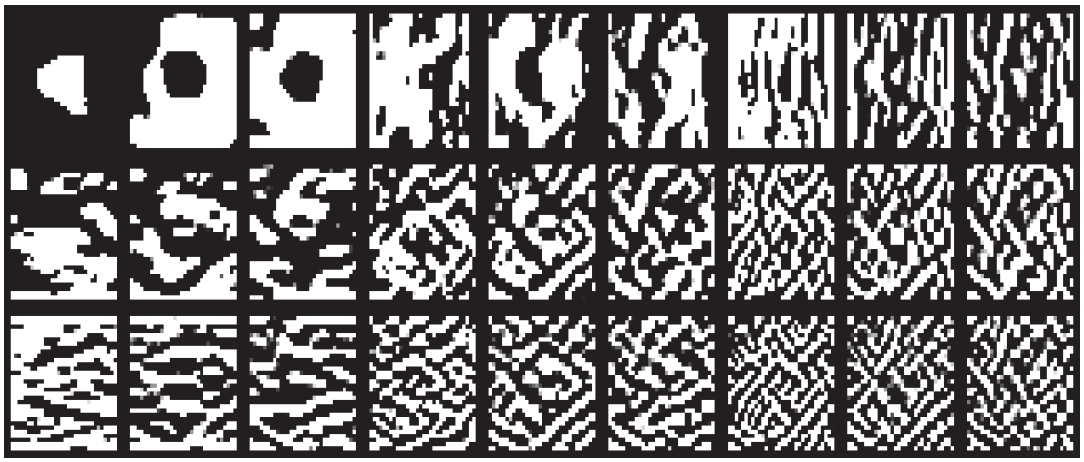
Despite having close resemblance between the feature extraction mechanism of the **PCANet** and the **TFNet**. These two networks capture visibly distinguishable features from the two view of the images as shown in Fig. 3.1. These plots are obtained by convolving image of a *cat* with the convolution filters obtained in the first layer of the networks. Undoubtedly, each of the L_1 outputs within the **PCANet** is visibly distinct. The outputs within the **TFNet** show visual similarity, i.e., the images in a triplet sequence show similarity consecutively. These plots demonstrate that the **TFNet** emphasizes mining the *common* information from the minutiae view of the data, whereas the **PCANet** emphasizes mining the *unique* information from the amalgamated view of the data. Both these kinds of information are proven beneficial for classification in [73, 118] and motivate the development of **HybridNet** described in the next section.

3.5 The Hybrid Network

The **PCANet** and the **TFNet** extract contrasting information from the amalgamated view and the minutiae view of the data, respectively. However, we hypothesize that the information from both of these views are essential as they conceal complementary information and that their integration can enhance the performance of classification systems. Motivated by the above, we propose the **HybridNet**, which simultaneously extracts information from both views of the data and is de-



(a) Convolution output from the **PCANet** where each output is visually distinct from the rest, depicting extraction of unique information with the amalgamated view of the data.



(b) Convolution output from the **TFNet** where the visual resemblance is observed in a sequence of three; depicting extraction of common information with minutiae view of the data.

Figure 3.1 : Comparison of convolution outputs from Layer1 in PCANet and TFNet on CIFAR-10 dataset. These plots demonstrate the contrast between the two types of information obtained with the two views of the data.

tailed in the next subsection. However for ease of understanding we illustrate the complete procedure of feature extraction with **Attn-HybridNet** in Fig. 3.2.

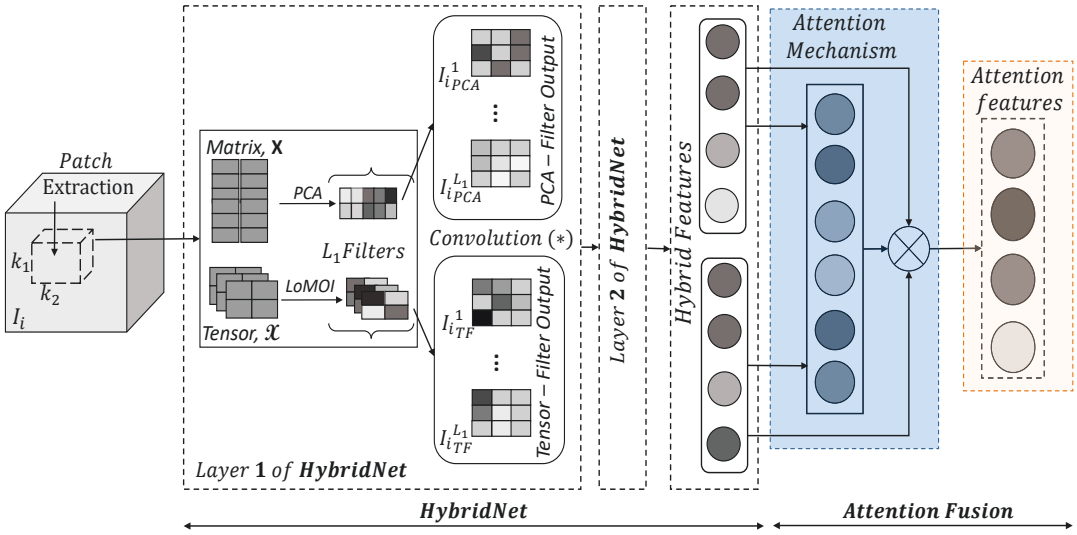


Figure 3.2 : Workflow of the **Attn-HybridNet** model.

3.5.1 The First Layer

Similar to the previous networks, we begin the feature extraction process by collecting all overlapping patches of size $k_1 \times k_2$ around each pixel from the image I_i. Importantly, the first layer of **HybridNet** consists of image-patches expressed both as tensors $\boldsymbol{\mathcal{X}} \in \mathbb{R}^{k_1 \times k_2 \times 3 \times N\tilde{m}\tilde{n}}$ and matrices $\boldsymbol{X} \in \mathbb{R}^{k_1 k_2 \times N\tilde{m}\tilde{n}}$ which are utilized for obtaining weights of convolution filters in layer 1 of **HybridNet**.

This enables this layer (and the subsequent layers) of **HybridNet** to learn superior filters as they perceive more information from both views of the data. The weights for the *pca*-filters are obtained as the principal-eigenvectors as $W^1_{l_{\mathbf{PCA}}} = mat_{k_1,k_2}(ql(\boldsymbol{X}\boldsymbol{X}^T))$, and the weights for convolution-tensor filters are obtained by utilizing *LoMOI* as $W^1_{l_{\mathbf{TF}}} = U^{(1)}_{(:,i)} \otimes U^{(2)}_{(:,j)} \otimes U^{(3)}_{(:,k)}$. Furthermore, the output from this layer is obtained by convolving input images with a) the PCA-filters and b) the convolution-tensorial filters in Eq. 3.15 and Eq. 3.16 respectively. This injects more diversity to the output from the first layer in **HybridNet** or equivalently to the

input of the succeeding layer of the **HybridNet**.

$$I^l_{i_{\mathbf{PCA}}} = I_i * W^1_{l_{\mathbf{PCA}}} \tag{3.15}$$

$$I^l_{i_{\mathbf{TF}}} = I_i * W^1_{l_{\mathbf{TF}}} \tag{3.16}$$

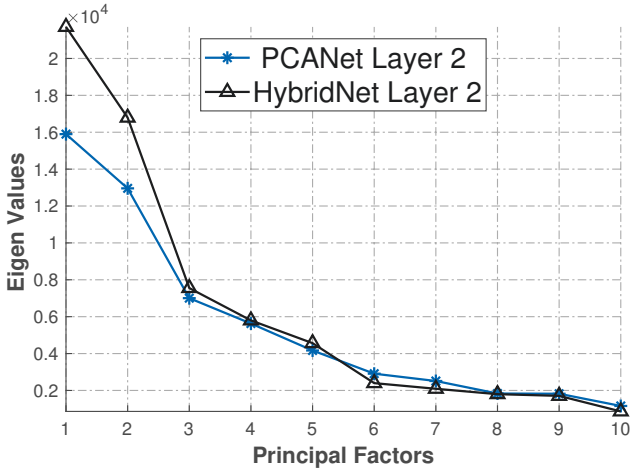
Since we obtain of L_1 *pca* filters and L_1 convolution-tensor filters, a total of $2 \times L_1$ outputs is obtained in this layer.

3.5.2 The Second Layer

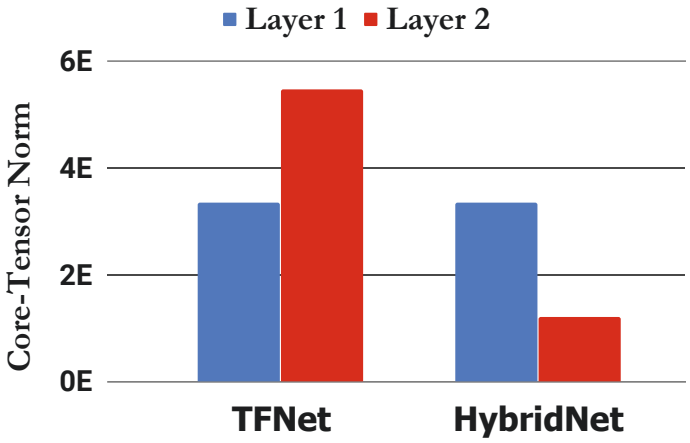
Similar to the first layer, we begin the process by collecting all overlapping patches of size $k_1 \times k_2$ around each pixel from the images. However, contrary to the above layer, the weights of the *pca*-filters $W^2_{l_{\mathbf{PCA}}}$ and convolution-tensor filters $W^2_{l_{\mathbf{TF}}}$ are learned from the data obtained by convolving input images with the *pca* filters and the convolution-tensor filters i.e. both $I_{i_{\mathbf{PCA}}}$ and $I_{i_{\mathbf{TF}}}$. Hence both the patch-matrix $\boldsymbol{Y} \in \mathbb{R}^{k_1 k_2 \times 2L_1 N\tilde{m}\tilde{n}}$ and the patch-tensor $\boldsymbol{\mathcal{Y}} \in \mathbb{R}^{k_1 \times k_2 \times 2L_1 N\tilde{m}\tilde{n}}$ contain image patches obtained from $[I_{i_{\mathbf{PCA}}}, I_{i_{\mathbf{TF}}}]$. This enables the hybrid filters to assimilate more variability present in the data while obtaining weights of their convolution filters. This phenomena is evident in Fig. 3.3.

The plot in Fig. 3.3(a) compares the eigenvalues obtained in layer 2 (we exclude eigenvalues from layer 1 as they completely overlap as their expected behavior). The leading eigenvalues obtained in layer 2 of the **HybridNet** by *principal components* have much higher magnitude than the corresponding eigenvalues obtained by *principal components* in **PCANet**. This demonstrates that the *pca* filter in the **HybridNet** captures more variability from the amalgamated view of data than the **PCANet**.

Similarly, Fig 3.3(b) compares the core-tensor strength in different layers of the **HybridNet** and the **TFNet**. We plot the norm of the core-tensor for both the networks as the values in the core-tensor are analogous to eigenvalues for higher-order



(a) Comparison of eigenvalues between networks



(b) Comparison of core-tensor strength between networks

Figure 3.3 : Comparison of factorization strength in Layer 2 of the **PCANet**, **TFNet** and **HybridNet** on CIFAR-10 dataset

matrices, and its norm signifies the compression strength of the factorization [75]. Again, the norm of the core-tensor in layer 2 of **HybridNet** is much lower than that of the **TFNet**, suggesting relatively higher factorization strength in **HybridNet**. Besides, as expected, the norm of the core-tensor in layer 1 for both the networks coincides and signifies equal factorization strength at this layer. Consequently, this leads to attainment of better-disentangled feature representations with the **HybridNet** and hence enhances its generalization performance over the **PCANet** and the

TFNet by integrating information from the two views of the data.

In the second layer, the weights of *pca* filters are obtained by *principal components* as $W^2_{l_{\mathbf{PCA}}} = mat_{k_1,k_2}(ql(\boldsymbol{Y}\boldsymbol{Y}^T))$ and the weights for convolution-tensor filters are obtained as $W^2_{l_{\mathbf{TF}}} = \mathbf{V}^{(1)}_{(:,i)} \otimes \mathbf{V}^{(2)}_{(:,j)}$, where the matrix factors are obtained using $LoMOI\ [\hat{\mathcal{Y}}, \mathbf{V}^{(1)}, \mathbf{V}^{(2)}] \leftarrow LoMOI(\mathcal{Y}, r_1, r_2)$. Analogous to the previous networks, the output images from this layer of **HybridNet** are obtained by a) convolving the L_1 images corresponding to the output from the PCA-filters in the first layer with the L_2 *pca* filters obtained in the second layer (Eq. 3.17), and b) convolving the L_1 images corresponding to the output from the convolution-tensorial filters in the first layer with the L_2 convolution-tensorial filters obtained in the second layer (Eq. 3.18). This generates a total of $2 \times L_1 \times L_2$ output images in this layer.

$$O^l_{i_{\mathbf{PCA}}} = I^l_{i_{\mathbf{PCA}}} * W^2_{l_{\mathbf{PCA}}} \tag{3.17}$$

$$O^l_{i_{\mathbf{TF}}} = I^l_{i_{\mathbf{TF}}} * W^2_{l_{\mathbf{TF}}} \tag{3.18}$$

The output images obtained from the *pca*-filters ($O^l_{i_{\mathbf{PCA}}}$) in layer 2 are then processed with the output layer of the **PCANet** (Sec. 3.3.1) to obtain $f_{i_{\mathbf{PCA}}}$ as the information from an amalgamated view of the image. Similarly, the output images obtained from the convolution-tensor filters ($O^l_{i_{\mathbf{TF}}}$) are processed to obtain $f_{i_{\mathbf{TF}}}$ as the information from minutiae view of the image. Finally, these two kinds information are concatenated to obtain the hybrid features as in Eq. 3.19.

$$f_{i_{\mathbf{hybrid}}} = [f_{i_{\mathbf{PCA}}}\ f_{i_{\mathbf{TF}}}] \in \mathbb{R}^{(2^{L_2})2L_1B} \tag{3.19}$$

The hybrid features obtained above couple the advantages of both the common and unique information obtained with the two views of the data. However, it still suffers from the feature redundancy induced by the spatial pooling operation in

the output layer. To alleviate this drawback, we propose **Attn-HybridNet**, which enhances the discriminability of hybrid features and is described in the next section.

3.6 Proposed attention-based fusion Attn-HybridNet

The proposed **HybridNet** eradicates the loss of information by integrating the learning scheme of **PCANet** and **TFNet** thus obtaining superior features than either of the networks. However, the feature encoding scheme in the output layer is elementary and induces redundancy in the feature representations [52, 23]. Moreover, the generalized spatial pooling operation in the output layer is unable to accommodate the spatial structure of the natural images, i.e., it is more effective for aligned images dataset like face and handwritten digits than for object recognition dataset. Simply, the design of the output layer is ineffectual to obtain utmost feature representation on object recognition datasets resulting in performance degradation with the **HybridNet**. Moreover, efficient techniques to alleviate this drawback with the output layer are not addressed in the literature, which necessitate the development of our proposed attention-based fusion scheme i.e. the **Attn-HybridNet**.

Our proposed attention-based fusion scheme is presented in Alg. 3, where $f_{\mathbf{hybrid}} \in \mathbb{R}^{N\times(2^{L_2})L_1B\times 2}$ are the hybrid feature vectors obtained with the **HybridNet**, $w \in \mathbb{R}^d$ is the feature level context vector of dimension $d << (2^{L_2})L_1B$, $\alpha^T \in \mathbb{R}^2$ is the normalized importance weight vector for combining the two kinds of information with attention fusion, and $F_{attn} \in \mathbb{R}^{(2^{L_2})L_1B}$ are the attention features. The fully connected layers i.e. $W \in \mathbb{R}^{d\times(2^{L_2})L_1B}$ and f_c are utilized to obtain hidden representations of features while performing attention fusion.

A few numerical optimization based techniques proposed in [52, 51] exist for alleviating the feature redundancy from architectures utilizing generalized spatial pooling layers. However, these techniques require grid search between the dictionary size (number of convolution filters in our case) and the pooling blocks in the output

Algorithm 3 Attn-HybridNet

Input: $f_{\mathbf{hybrid}} = [f_{\mathbf{PCA}};\ f_{\mathbf{TF}}] \in \mathbb{R}^{N \times (2^{L_2})L_1 B \times 2}$ the hybrid feature vectors from the training images; $\boldsymbol{Y} = [0,\ 1, ...,\ C]$ ground truth of training images, dimensionality of feature level attention context vector $w \in \mathbb{R}^d$ where $d << \mathbb{R}^{(2^{L_2})L_1 B}$.

randomly initialize W, f_c, and w

$loss \leftarrow 1000$ ▷ arbitrary number to start training

do

  $[f_{batch}, \boldsymbol{Y}_{batch}] \leftarrow$ sample batch $([f_{\mathbf{hybrid}}, \boldsymbol{Y}])$

  $P_F \leftarrow tanh(W.f_{batch})$ ▷ get the hidden representation of the hybrid features

  $\alpha = softmax(w^T.P_F)$ ▷ measure and normalize the importance

  $F_{attn} = f_{batch}.\alpha^T$ ▷ perform attention fusion

  $\hat{\boldsymbol{Y}} \leftarrow f_c(F_{attn})$ ▷ fully connected layer

  $loss \leftarrow LogLoss(\boldsymbol{Y}_{batch}, \hat{\boldsymbol{Y}}_{batch})$ ▷ compute loss for optimizing parameters

  back-propagate loss for optimizing W, f_c, and w.

while $[(loss \geq \varepsilon)]$ ▷ loop until convergence

Output: parameters to perform attention fusion W, f_c, and $w \in \mathbb{R}^d$

layer while performing optimization. Besides, the transition to prune filters from a single-layer networks to multi-layer network is not smooth in these techniques. A major difference between our proposed **Attn-HybridNet** and the existing proposal in [52, 51] is that we reduce the feature redundancy by performing feature selection with attention-based fusion scheme, whereas the existing techniques prune the filters to eliminate the feature redundancy. Therefore, our proposed **Attn-HybridNet** is superior to these existing techniques as it decouples the two sub-processes, i.e., information discovery with convolution layers and feature aggregation in the pooling layer while alleviating the redundancy exhibiting in the feature representations.

The discriminative features obtained by **Attn-HybridNet** i.e. F_{attn} are utilized with $softmax$-layer for classification, where the parameters in the proposed fusion scheme (i.e., W, f_c and w) are optimized via gradient-descent on the classification loss. This simple yet effective scheme substantially enhances the classification performance by obtaining highly discriminative features. Comprehensive experiments are conducted in this regard to demonstrate the superiority of **Attn-HybridNet**

detailed in Sec. 3.7.

3.6.1 Computational Complexity

To obtain the computational complexity of **Attn-HybridNet**, assume we assume attention-based fusion scheme with two-layer **HybridNet** with a patch size of $k_1 = k_2 = k$.

In each layer of the **HybridNet**, we have to compute the time complexities arising from learning convolution weights from the two views of the data. The formation of the zero-centered patch-matrix X and zero-centered patch-tensor $\mathcal{X}$ has identical complexities as $k^2(1 + \tilde{m}\tilde{n})$. The complexity of eigen-decomposition for patch-matrix and tensor factorization with *LoMOI* for patch-tensor are also identical and equal to $\mathcal{O}((k^2)^3)$, where k is a whole number < 7 in our experiments. Further, the complexity for convolving images with the convolution filters at stage i requires $L_i k^2 mn$ flops. The conversion of L_2 binary bits to a decimal number in the output layer costs $2L_2\tilde{m}\tilde{n}$, where $\tilde{m} = m - \lceil \frac{k}{2} \rceil, \tilde{n} = n - \lceil \frac{k}{2} \rceil$ and the naive histogram operation for this conversion results in complexity equal to $\mathcal{O}(mnBL_2log2)$.

The complexity of performing matrix multiplication in **Attn-HybridNet** is $\mathcal{O}\Big(2L_1B\big(d(1 + 2^{L_2}) + 2^{L_2}\big)\Big)$ which can be efficiently handled with modern deep learning packages like Tensorflow [1] for stochastic updates. To optimize the parameters in the attention-based fusion scheme (W, f_c, and w), we back-propagate the loss through the attention network until convergence of the error on the training features.

3.7 Experiments and Results

3.7.1 Experimental Setup

In our experiments, we utilized two-layer architecture for each network while obtaining weights of their convolution filters. The number of convolution filters in

the first and the second layer are optimized via cross-validation on each dataset. The dimensionality of the feature vectors extracted from **PCANet** and **TFNet** becomes $BL_1 2^{L_2}$, assuming L_1 and L_2 as the number of convolution filers in layer 1 and layer 2 respectively. The dimensionality of feature vector with **HybridNet** becomes $2BL_1 2^{L_2}$. We utilized *Linear-SVM* [31] as the classifier with features obtained with the **PCANet**, **TFNet**, and the **HybridNet**.

The attention-based fusion scheme is performed by following the procedure as described in Alg. 3 where, we searched the optimal attention dimension for the context level feature vector $w \in \mathbb{R}^d$ in $[10, 50, 100, 150, 200, 400]$. The obtained attention features i.e. $F_{attn} \in R^{BL_1 2^{L_2}}$ are utilized with $softmax$-layer for classification. The parameters of attention-based fusion scheme (W, f_c, and w) are optimized via back-propagation on the classification loss implemented in TensorFlow [1]. We observed that the attention-network's optimization took less than 15 epochs for convergence on all the datasets.

Furthermore, in our experiments, we do not utilize any data-augmentation techniques like rotations, random cropping, etc. to increase the size of the training data.

3.7.2 Datasets

We utilize the following datasets and hyper-parameters in our experiments:

1 MNIST variations [68], which consists of 28×28 gray scale handwritten digits with controlled factors of variations such as background noise, rotations, etc. Each variation contains $10K$ training and $50K$ testing images. We cite the results for state of the art techniques like 2-stage ScatNet [10] (ScatNet-2) and 2-stage Contractive auto-encoders [99] (CAE-2) as published in [13] while comparing the performance of our proposed **Attn-HybridNet** as baselines.

The parameters of **HybridNet** (and other networks) are set as $L_1 = 9$, $L_2 = 8$, $k_1 = k_2 = 7$, with block size $B = 7 \times 7$ and size of overlapping regions between the blocks equal to half of the block size while performing feature pooling with all the networks.

2 CUReT texture dataset [116], consists images with dimensions 200×200 for 61 texture categories, where each category has images of the same material with different pose, illumination conditions, specularity, shadowing, and surface normals. Following the standard procedure in [116, 13] a subset of 92 cropped images were taken from each category and randomly partitioned into train and test sets with a split ratio of 50%. The classification results are averaged over 10 different trails. We set, $L_1 = 9$, $L_2 = 8$, $k_1 = k_2 = 5$, with the block size $B = 50 \times 50$ and the size of overlapping regions between the blocks is equal to half of the block size. Again, we cite the results of the baselines techniques as published in [13].

3 CIFAR-10 [65] dataset consists of RGB images of dimensions 32×32 for object recognition consisting of $50K$ and $10K$ images for training and testing respectively. These images are distributed among 10 classes and vary significantly in object position, object scale, colors, and textures but also within each class. We varied the number of filters in layer 1 i.e., L_1 as 9 and 27 and kept the number of filters in layer 2 i.e. $L_2 = 8$. The patch-size k_1 and k_2 are kept equal and varied as 5, 7, and 9 with block size $B = 8 \times 8$. Following [13] we also applied spatial pyramid pooling (SPP) [36] to the output layer of **HybridNet** (and similarly to the out layer of other networks). We additionally applied PCA to reduce the dimension of each pooled feature to 100.[§]. These features are utilized with Linear-SVM for classification and **Attn-HybridNet** for obtaining

[§]Results does not vary significantly on increasing the projection dimensions.

attention features F_{attn}.

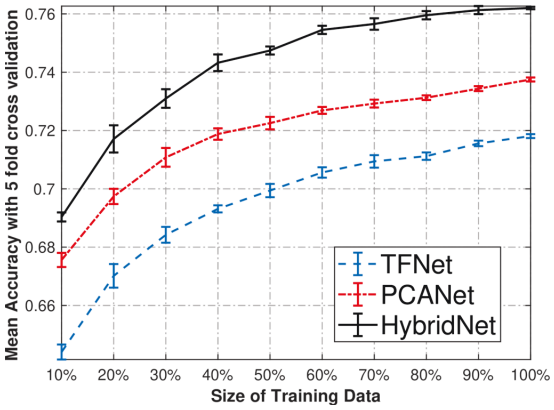
Classification accuracies from comparable methods such as Tiled CNN [85], CUDA-Convnet [64], VGG style CNN on CIFAR-10 (VGG-CIFAR10 reported by [18]), and K-means (tri) [22] are taken from their respective publications. However, for a qualitative case study, we have utilized publicly available source codes of these baselines and executed them by varying the size of training dataset. Importantly, we do not compare our method with complex architectures such as ResNet [42] and DenseNet [48] as these have sophisticated convolution operations but are the current state of the art on this dataset.

3.8 Results and Discussions

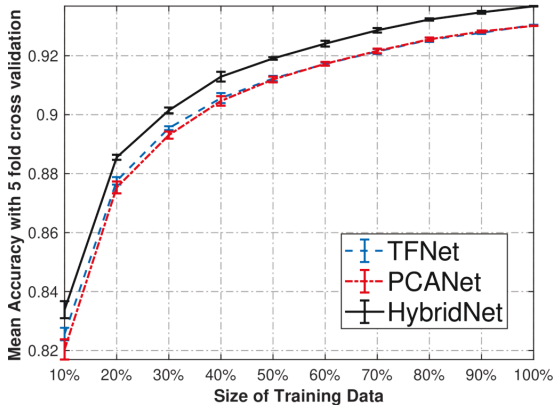
The main contributions in this chapter are 1) the integration of information available from both the amalgamated view (i.e., the unique information) and the minutiae view (i.e., the common information), and 2) the attention based fusion of information obtained from the two views of the data for supervised classification. We evaluate the significance of these contributions under the following research questions:

Q1: Is the integration of both the minutiae view and the amalgamated view beneficial? Or, does their integration deteriorate the generalization performance of **HybridNet**?

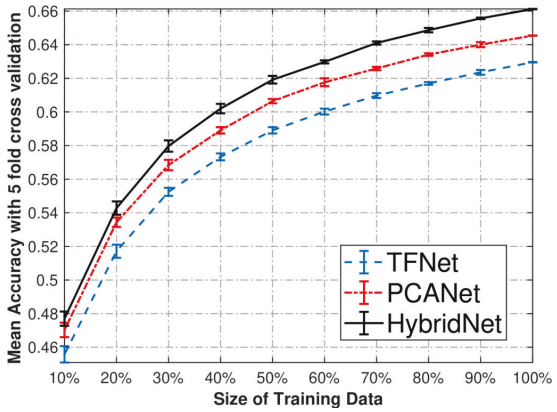
To validate this, we evaluate the classification performance of **HybridNet**, the **PCANet**, and **TFNet** on CIFAR-10 and MNIST variations datasets by varying the amount of training dataset while extracting feature. We then obtained classification accuracies of these features on the test dataset, and plot the mean and variance with 5 fold cross-validation in Fig. 4.3.



(a) CIFAR-10



(b) MNIST Rotation



(c) MNIST bg-img-rot

Figure 3.4 : Performance Comparison by varying size of the training data

These plots suggest that the classification accuracy obtained with the features from **HybridNet** (and also from the **PCANet** and the **TFNet**) linearly increases with respect to the size of training data. Moreover, these plots also suggest that the information obtained from the amalgamated view in **PCANet** is superior than the information obtained from the minutiae view **TFNet** on object-recognition dataset. However, these two kinds of information achieve competitive classification performance on variations of handwritten digits dataset which contains nearly aligned images.

Most importantly, these plots unambiguously demonstrate that integrating both

Parameters				**PCANet**	**TFNet**	**HybridNet**	**Attn-HybridNet**
L_1	L_2	k_1	k_2	Error (%)	Error (%)	Error (%)	Error (%)
8	8	5	5	34.80	32.57	31.39	**28.08**
8	8	7	7	39.92	37.19	35.24	**30.94**
8	8	9	9	43.91	39.65	38.04	**35.33**
27	8	5	5	26.43	29.25	23.84	**18.41**
27	8	7	7	30.08	32.57	28.53	**25.67**
27	8	9	9	33.94	34.79	31.36	**27.70**

Table 3.2 : Classification Error (%) obtained by varying hyper-parameters on CIFAR-10 dataset without augmentation

these information obtain superior feature representations, consequently improving the classification performance of the proposed **HybridNet**.

Q2: How does the hyper-parameters affect the performance of the **HybridNet**. Moreover, what is the effect of these features representation on the proposed attention based fusion scheme?

To address this question, we present a detailed study on how the hyper-parameters affect the performance of **HybridNet** and **Attn-HybridNet**. In this regard, we compare the classification performance of the **PCANet**, **TFNet**, **HybridNet**, and **Attn-HybridNet** on CIFAR-10 dataset in Table. 3.2. The lowest error is highlighted in slightly larger font, while the minimum error achieved in each row is highlighted in bold font. Moreover, we also illustrate the performance of **Attn-HybridNet** by varying the dimension of context level feature vector w utilized in our attention-fusion scheme in Fig. 3.5.

A clear trend is visible in Table. 3.2 among the performances of all the networks, where the classification error decreases with an increase in the number of filters in the first layer of the networks. This trend also demonstrates the effect of the factorization rank while obtaining the *principal-components* and the matrix factors with *LoMOI*; signifying that increasing the number filters in the first layer allows all the networks to increase the data variability that aids in obtaining better feature correspondences in the output stage. In addition, this also increases the dimensionality of the features extracted by the networks suggesting that comparatively higher dimensional features have lower intraclass variability among the feature representations of objects from the same category.

Another trend can be observed in the performance table where the classification error increases with the patch size. Since the dimension of images in CIFAR-10 is 32×32, this may be due to the presence of less background with smaller image-patches as increasing the patch size gradually mount to non-stationary data [13].

More importantly, our proposed **Attn-HybridNet** substantially reduces the classification error by **22.78**%, when compared to classification performance with **HybridNet** on CIFAR-10 dataset. The plot in Fig. 3.5 shows the effect on classification accuracy by varying dimensions of feature level context vector w in **Attn-HybridNet**.

Q3: What is the performance of **Attn-HybridNet** (and **HybridNet**) compared to other popular baseline techniques?

We compare the performance of popular (neural and non-neural) methods which are comparable in architecture and learning scheme to the proposed **Attn-HybridNet** and **HybridNet** on CIFAR-10, MNIST variations, and CUReT datasets in Table. 3.3 and Table. 3.4 respectively. Furthermore, to perform qualitative analysis we plot the classification performance of different schemes on CIFAR-10 dataset

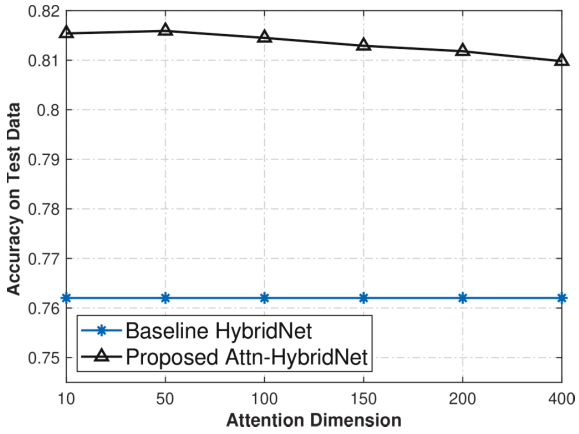


Figure 3.5 : Accuracy of Attn-HybridNet on CIFAR-10 dataset by varying the dimension of attention context vector w in Alg. 3.

in Fig. 3.6 and visualize the feature embeddings from **HybridNet** and **Attn-HybridNet** with t-SNE plot [78] in Fig. 3.7.

On MNIST handwritten digits variations dataset, the **Attn-HybridNet** (and also the **HybridNet**) outperforms the state of the art results on five out of seven variations. In particular, for *bg-rand* and *bg-img* variations, we decreased the error (compared to [119]) by **31.68**% and **13.80**% respectively.

On CUReT texture classification dataset, the **Attn-HybridNet** achieves the lowest classification error among all the networks, albeit it achieves slightly higher classification error compared to state of the art. The difference in classification error achieved by state of the art [10] and **Attn-HybridNet** is marginal and is only 0.5%.

On CIFAR-10 object recognition dataset, we present multiple qualitative case studies and quantitative performance measurements between our proposed **Attn-HybridNet** and other baseline schemes.

Qualitative Discussion We discuss the insights of performance achieved by various schemes and our proposal by plotting the classification performance achieved

Methods	baisc	rot	bg-rand	bg-img	bg-img-rot	rect-image	convex
CAE-2 [99]	2.48	9.66	10.90	15.50	45.23	21.54	-
TIRBM [106]	-	**4.20**	-	-	35.50	-	-
PGBM [107]	-	-	6.08	12.25	36.76	**8.02**	-
ScatNet-2 [10]	1.27	7.48	12.30	18.40	50.48	15.94	6.50
PCANet	1.07	6.88	6.99	11.16	35.46	13.59	4.15
TFNet	1.07	7.15	6.96	11.44	37.02	16.87	4.98
HybridNet	1.01	6.32	5.46	10.08	33.87	12.91	3.55
Attn-HybridNet	**0.94**	4.31	**3.73**	**8.68**	**31.33**	10.65	**2.81**

Table 3.3 : Classification Error (%) obtained on MNIST variations datasets

by varying the size of training data in Fig. 3.6. Although our proposed **Attn-HybridNet** consistently achieved the highest classification performance, few interesting trends are still noticeable.

The first trend is regarding the lower classification performance achieved by both CUDA-Convnet [64] and VGG-CIFAR10 [53] with less amount of training dataset, particularly until 40%. It is intuitive and justifiable since less amount of the training data is not sufficient to efficiently learn the parameters of these deep networks. However, on increasing the amount of training data (above 50%), the performance of these networks increases substantially i.e., increases with a larger margin compared to the performance of SVM based schemes in **HybridNet** and K-means (tri) [22].

A second trend can be noticed with the classification performances of **HybridNet** and K-means (tri). Both, these networks achieve higher classification accuracy compared to the deep networks with less amount of training data; particularly the **HybridNet** has **11.56**% higher classification rate compared to the second high-

Methods	Error (%)
Textons [40]	1.50
BIF [25]	1.40
Histogram [9]	1.00
ScatNet [10]	**0.20**
PCANet	0.84
TFNet	0.96
HybridNet	0.81
Attn-HybridNet	0.72

Table 3.4 : Classification Error (%) obtained on CUReT datasets

est classification accuracy achieved by K-means (tri) with only 10% of the training dataset. However, the accuracy of these networks does not scale or increase substantially with an increase in the training data, as noticed with the deep-network based schemes.

Lastly, the **Attn-HybridNet** achieves the highest classification performance among all the techniques with any subset of the training dataset. One possible reason can be the requirement of fewer parameters with proposed attention-fusion while performing feature selection with attention-based fusion for alleviating the feature redundancy.

Moreover, the t-SNE plot in Fig. 3.7 compares the discriminability of features obtained with the **HybridNet** and **Attn-HybridNet**. The plot on the features obtained from **Attn-HybridNet** Fig. 3.7(b) visually achieves better clustering than the plot on features obtained from **HybridNet** Fig. 3.7(a) and justifies the performance improvement with our proposal.

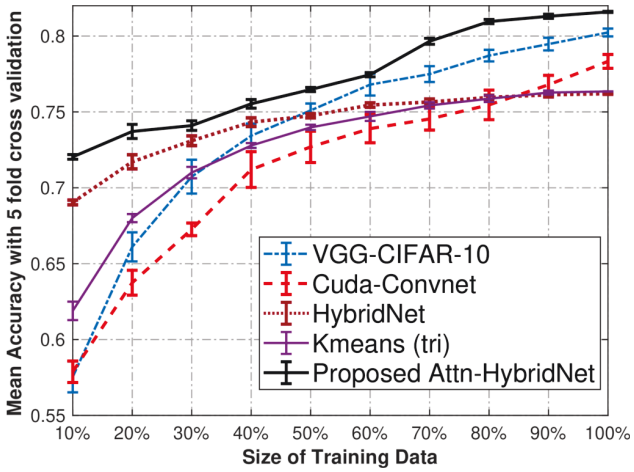


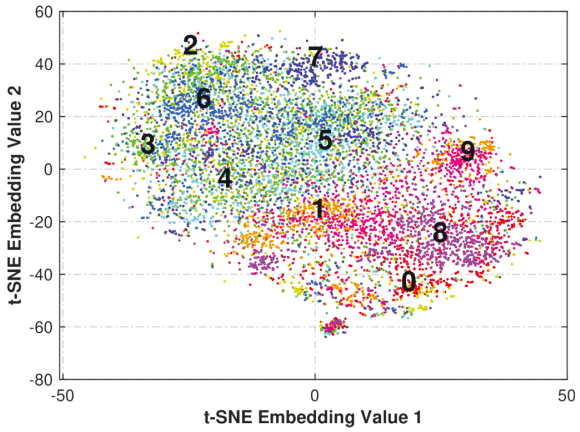
Figure 3.6 : (Best viewed in color) Accuracy of various methods on CIFAR-10 dataset by varying size of the training data

Quantitative Performance Our proposed **Attn-HybridNet** achieves much lower error compared to Titled CNN [85], K-means (tri), and the **PCANet**; particularly 16.70% lower than K-means (tri) which has 2× higher feature dimensionality than our proposed **HybrdidNet** and utilizes L_2 regularized-SVM instead of *Linear-SVM* for classification.

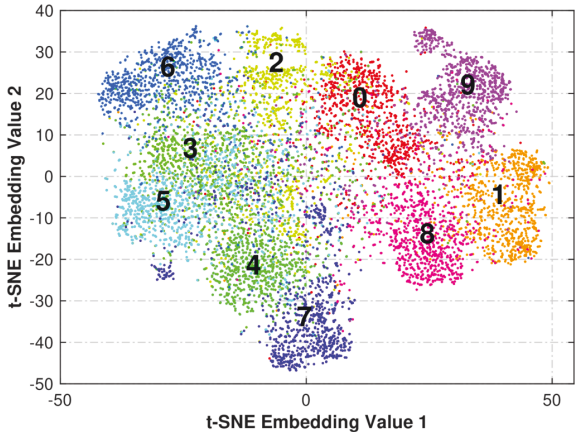
The performance of our proposed **Attn-HybridNet** is still better than VGG-CIFAR-10 [18] and comparable to CUDA-Convnet [64][¶], both of which have more depth than the proposed **Attn-HybridNet**. In particular, we have reduced the error by 1.63% than VGG-CIFAR-10 with 99.63% less trainable parameters. At the same time, we have performed very competitive to CUDA-Convnet achieving 0.41% higher error rate but with 88% less number of tunable parameters.

Nevertheless, the effort required to estimate the tuneable parameters (like convolution kernel size, number of convolution filters, etc.) with **Attn-HybridNet** is very convenient, and also the amount of training time and parameter size of our

[¶]we cite the accuracy as published and not from the qualitative analysis



(a) **HybridNet**



(b) **Attn-HybridNet**

Figure 3.7 : (Best viewed in color) t-SNE visualization of features from **HybridNet** (top) and **Attn-HybridNet** (bottom) on CIFAR-10 dataset.

proposed technique compared to the baseline deep-networks is negligible. Hence, the classification accuracy obtained with **Attn-HybridNet** is justifiable and encouraging.

Methods	**#Depth**	**#Params**	**Error**
Tiled CNN [85]	-	-	26.90
K-means (tri.) [22] (1600 dim.)	1	5	22.10
CUDA-Convnet [64]	4	1.06M	**18.00**
VGG-CIFAR-10 [53]	5	3.45M	20.04
PCANet	3	7	26.43
TFNet	3	7	29.25
HybridNet	3	7	23.84
Attn-HybridNet (proposed)	3	12.7k	18.41

Table 3.5 : Classification Error (%) obtained on CIFAR-10 dataset without data augmentation

3.9 Summary

The main focus of this chapter is to investigate the plausibility of building lightweight convolution neural networks that are independent of high performance architecture. We introduced **HybridNet**, which integrates the information discovery and feature extraction procedure from the amalgamated view and the minutiae view of the data. The development of **HybridNet** is motivated by the fact that information obtained from the two views of the data are individually insufficient but necessary for classification. To extract features from the minutiae view of the data, we proposed the **TFNet** that obtains weights of its convolution-tensor filters by utilizing our custom-built *LoMOI* factorization algorithm. We then demonstrated how the information obtained with the two views of data are complementary to each other. Then, we provided details to simultaneously extract the common information from the amalgamated view and unique information with the minutiae view of the data in our proposed **HybridNet**.

We then proposed **Attn-HybridNet** for alleviating the feature redundancy by performing attentive feature selection. Our proposed **Attn-HybridNet** enhances the discriminability of features, which further enhances their classification performance. The significance of our proposed **Attn-HybridNet** and **HybridNet** is demonstrated by classification performance on multiple real-world datasets.

Chapter 4

DeepCU: Integrating both Common and Unique Latent Information for Multimodal Sentiment Analysis

4.1 Introduction

Recent developments in Deep Learning techniques has led tremendous success in Sentiment Analysis and emotion recognition [126, 5, 83]. Despite the recent efforts in text processing for sentiment analysis in [77, 34, 122, 121], a core research challenge for this domain is the efficient utilization of multimodal representations such as voice and visual gestures for sentiment prediction [67, 132]. There is a growing trend of sharing opinion videos on social media platforms (Facebook, YouTube, etc.) which comprise of language (spoken words), visual-gestures, and acoustic (voice) as multimodal representations. Combining the unimodal representation for sentiment analysis becomes crucial as the combined information from multiple modalities promises better generalization capabilities over traditional text-based schemes [5, 92]. Figure 4.1 illustrates a typical multimodal sentiment analysis systems, where the utterance "That's – that's true" is ambiguous and can be perceived as positive or neutral sentiment. However, combining speaker's visual gesture which is 'smile' and loud-pitch of the acoustic in the video helps us in identifying the sentiment of the speaker.

Some recent promising attempts in combining multimodal representations are presented in [32, 129, 47, 74]. In all these fusion schemes, an outer product is taken among unimodalities to obtain the joint representation in the form of a tensor. In

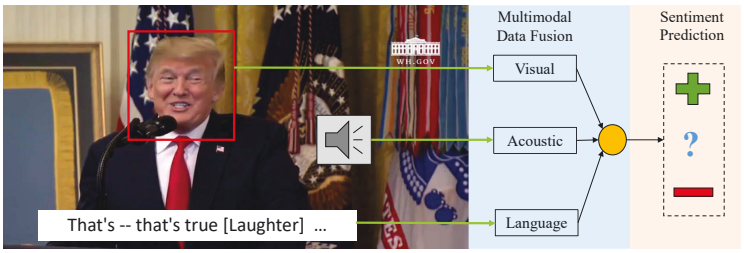


Figure 4.1 : A typical Multimodal Sentiment Analysis System

[32, 47] the authors only utilized the combined information offered by the tensor for multimodal data fusion. Whereas in [129, 74] the authors supplemented the information from the unimodality with the combined information from the tensor for multimodal data fusion. The above techniques either train a deep feed-forward neural network on the tensor or obtain its low-rank representation for multimodal fusion.

Although the fusion of interacting modalities i.e. acoustic, visual, and language often improves the generalization performance, there are various scenarios with real-world datasets which must be handled properly while performing fusion, otherwise the joint representation might become futile. A common scenario in this regard is the occurrence of missing values in the unimodal representations [67] which leads to futile joint representations. For visual features missing values can occur due to several reasons for example poor lighting in the opinionated video, the speaker is wearing accessories (hat, glasses etc.) or covers his face while laughing. Similarly, for the auditory signal factors like voice-echo, ambient noise can cause missing values in the feature set. Figure 4.2, illustrates a motivating example presenting limitations with the current state of the art fusion techniques i.e. TFN [129] shown as A., LMF [74] shown as B.; and superiority of our proposed **DeepCU** shown as C. in Figure 4.2 when faced with missing values.

In Figure 4.2, to obtain the joint representation from acoustic and language

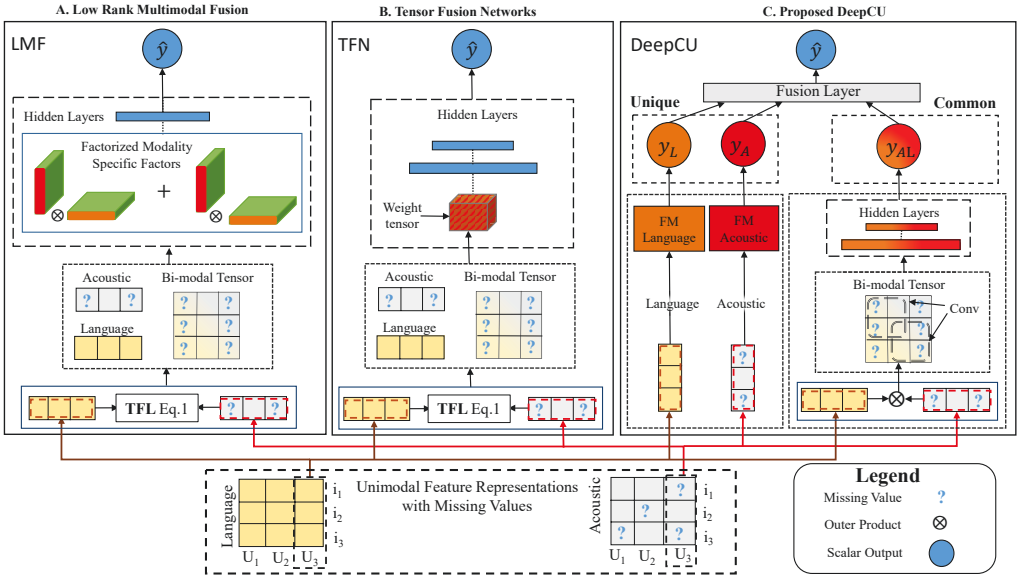


Figure 4.2 : Comparison of missing values (interrogation mark '?') scenarios by State of the art A. Low-rank Multimodal Fusion (LMF), B. Tensor Fusion Networks (TFN), and C. our Proposed DeepCU.

modalities the TFN and the LMF utilizes an outer product on the augmented features. This results in both the bi-modal and the unimodal features in joint representation (as tensor). However, the joint representation in all cases is much sparse (contains more missing values) than the acoustic modality and the learning mechanisms of both the TFN and LMF fail to efficiently extract information in this scenario. Our proposed **DeepCU** can handle the missing value scenario due to the following:

1. The convolution kernels split the joint representation into overlapping segments while performing feature extraction which reduces the impact of missing values.

2. Factorization Machines (FMs) obtaining modality-specific unique information are robust with sparse feature vectors which subsides the impact on **DeepCU**'s

performance and information discovery when the joint representation is futile.

3. Learning unshared latent representation for common and unique networks ensures that latent-embeddings of the superior representations remain unaffected by influences of inferior representations (i.e. gradient from futile representation). This restriction enforces latent-embeddings to attain complementary information and provides more expressiveness while performing fusion in the higher layers.

Motivated by the above points, we propose a novel deep common and unique feature extraction technique for multimodal data fusion, which we call as **DeepCU**. Our proposed **DeepCU** has two components 1) unique sub-network which obtains information specific to individual modalities and; 2) common sub-network which obtains combined information from joint (multi-mode) representations by using proposed deep-convolution tensor networks. Information from the common and the unique sub-networks is integrated by a fusion layer to obtain an integrated output.

4.2 Our Contributions

The main contribution of this chapter are as follows:

I. We design a consolidated deep network for joint utilization and discovery of both the common (multi-mode) and unique (mode-specific) properties of the multimodal data for sentiment analysis.

II. Our proposed **DeepCU** is conceptually more expressive than existing state of the art (TFN and LMF) as it captures non-linear multi-mode interactions exhibiting in the tensorial representation within our common network sub-network. Moreover, our unique sub-network obtains both linear and factorized non-linear (quadratic) feature relations which mitigates the missing value scenarios and enhances the generalization capability of **DeepCU**.

III. We perform comprehensive experiments on multimodal CMU-MOSI and POM datasets and demonstrate the effectiveness of utilizing both common and unique latent information with comparisons to other techniques.

4.3 Related Work

We focus our review on recent neural based frameworks for multimodal data fusion proposed in the literature. In [71] a bilinear-CNN is proposed to obtain bi-modal interactions among features obtained from two heterogeneous CNNs. However, the bilinear layer required parameter estimation of a quadratic number of neurons, and hence prone to over-fitting. This limitation is alleviated in [32] which introduces an alternate formulation of the bilinear layer and obtains its compact representation by utilizing sophisticated neural based factorization schemes.

However, the above fusion schemes only express the bi-modal (or tri-modal) interactions from unimodal representations either as: a) inter-modal (outer product) or b) intra-model (simple concatenation) based representations. But utilization of both the intra-modal and inter-modal representations are proven helpful in many machine learning tasks [73, 118, 119]. In this regard, Tensor Fusion Layer (TFL) is proposed in [129] which leverages the expressiveness of both the inter-model and the intra-model fusion schemes.

The TFL applies bilinear product by augmenting the unimodal representations with an additional feature of constant values equal to 1. The outer product on the augmented unimodal representations now yields two sets of information: 1) the bi-modal (or tri-modal) interactions in the form of 2D-tensor (3D-tensor) and 2) the raw unimodal representations of the modalities. Mathematically the TFL for bi-modal interactions can be expressed as in Equation (4.1), where $\boldsymbol{x}_1 \in \mathbb{R}^n$ and

$\boldsymbol{x}_2 \in \mathbb{R}^m$ are feature vectors from two different modalities

$$TFL(\boldsymbol{x}_1, \boldsymbol{x}_2) = \boldsymbol{\mathcal{X}} = \begin{bmatrix} \boldsymbol{x}_1 \\ 1 \end{bmatrix} \otimes \begin{bmatrix} \boldsymbol{x}_2 \\ 1 \end{bmatrix} = \begin{bmatrix} \boldsymbol{x}_1 \otimes \boldsymbol{x}_2 & \boldsymbol{x}_1 \\ \boldsymbol{x}_2 & 1 \end{bmatrix} \tag{4.1}$$

'$\otimes$' represents the outer product and $\boldsymbol{\mathcal{X}} \in \mathbb{R}^{(n+1)\times(m+1)}$.

4.3.1 Tensor Fusion Networks (TFN)

The TFN proposed in [129] learns a weight tensor $\boldsymbol{\mathcal{W}} \in \mathbb{R}^{(n+1)\times(m+1)\times k}$ and a set of feed-forward layers to obtain the combined information from $\boldsymbol{\mathcal{X}}$. The TFN outperformed all the previous fusion schemes for multimodal sentiment analysis on CMU-MOSI dataset as it leverages the expressiveness offered by both the bimodal and unimodal information exhibiting in the joint representations obtained via TFL. However, the dimensionality of the weight tensor $\boldsymbol{\mathcal{W}}$ increases exponentially by increasing the number of unimodal representations for fusion and hence the TFN is not scalable [74].

4.3.2 Low-rank Multimodal Fusion (LMF)

The LMF proposed in [74] alleviates the scalability issues with TFN by approximating lower dimensional modality specific factors (commonly refereed as Rank-k tensors in CP decomposition [73]). The LMF, the weight tensor $\boldsymbol{\mathcal{W}}$ is equivalently expressed as $\boldsymbol{\mathcal{W}} \equiv \left(\boldsymbol{W}_1 \otimes \boldsymbol{W}_2\right)$, where $\boldsymbol{W}_1 \in \mathbb{R}^{(n+1)\times k}$, $\boldsymbol{W}_2 \in \mathbb{R}^{(m+1)\times k}$. Extracting of information from $\boldsymbol{\mathcal{X}}$ is now reformulated as: $\left(\hat{\boldsymbol{x}}_1 \times \boldsymbol{W}_1\right) \odot \left(\hat{\boldsymbol{x}}_2 \times \boldsymbol{W}_2\right)$, where $\hat{\boldsymbol{x}}_i = \begin{bmatrix} \boldsymbol{x}_i, 1 \end{bmatrix}^T$; and '$\odot$' is the element-wise product operator; and '$\times_i$' is the mode-i product between tensor and matrix. Hence, explicitly learning higher dimension weight tensor $\boldsymbol{\mathcal{W}}$ with TFN is not required. The LMF is a current state of the art on CMU-MOSI dataset without any contextual information.

4.3.3 Hybrid - DeepShallow

$Hybrid_{DS}$ [124] network is the most recent work on multimodal data fusion for sentiment analysis. $Hybrid_{DS}$ first trains deep networks on visual and acoustic features independently and concatenates the final layers of these networks to obtain the combined representation. Further, to extract information from the language modality, it trains a SVM classifier (as shallow network) on the language features. Finally, Random Forest is trained on the predicted value from SVM and the combined representation from the deep network for multimodal sentiment prediction. However, the $Hybrid_{DS}$ is not proposed on CMU-MOSI dataset but for this work, it is justifiable to be included as a baseline. Similar to TFN and LMF, the $Hybrid_{DS}$ network also falls prey to the missing values (as discussed in Section 4.1) while performing multimodal fusion.

Approaches like [131, 92, 130] incorporates contextual information from multimodal representations utilize an attention mechanism to incorporate the information available from all utterances of the same speaker which enables them to model the complex dynamics of inter-modality relationships efficiently. Although these techniques are superior than the above schemes, they require additional information like the identity of the speaker, the sequence of the utterance-sentiments while modelling their fusion schemes. This additional information might not be available in the general scenarios.

4.4 Proposed Methodology

Contrary to the existing fusion schemes we aim to utilize both the common and the unique information for multimodal data fusion. To this end, we first propose two sub-networks, i.e., 1) unique network for obtaining modality-specific features (described in Section 4.4.1) and; 2) common network which consists of proposed deep-convolution tensor networks (described in Section 4.4.2). The latent space for

Fusion Schemes	Deep & Shallow	Inter Modality	Modality Specific	Convolution	Unshared Embeddings
DeepFM [38]	✓	×	×	×	×
TFN [129]	×	✓	✓	×	×
LMF [74]	×	✓	✓	×	×
Hybrid$_{DS}$ [124]	×	×	✓	×	×
DeepCU (proposed)	✓	✓	✓	✓	✓

Table 4.1 : Comparison of multimodal data fusion models

the unique information and the common information is unshared (i.e. influenced only by gradient of their respective sub-network) and allows **DeepCU** to obtain complementary information with both the sub-networks. Later, these two kinds of information are integrated via a fusion layer (described in Section 4.4.3 which allows joint optimization and information discovery in common and unique network's) to $\hat{y}$ as the final prediction from **DeepCU**. The differences and similarities between existing multimodal data fusion techniques and the proposed **DeepCU** are summarized in Section 4.4.

The raw feature vectors from a single utterance for acoustic and visual modalities are denoted as $z_a \in \mathbb{R}^{1\times k_a}$ and $z_v \in \mathbb{R}^{1\times k_v}$ respectively, where k_a and k_v represent the dimensionality of the feature vectors. For language modality the raw features are word-embeddings denoted as $z_l \in \mathbb{R}^{1\times s_l\times d_l}$, where s_l is the sequence length of the embeddings and k_l is the dimensionality of each sequence vector. The latent space (or embeddings) obtained from these features for the common and unique sub-networks are unshared and influenced only by their respective networks. This restriction allows both networks to learn complementary feature representations at lower layers which enhances their expressiveness in the fusion layer. Besides, optimizing unshared latent space is empirically shown beneficial in [43].

4.4.1 Unique Network

The modality-specific information is obtained by utilizing Factorization Machines (FMs). There are two main motivations behind utilizing FMs instead of any other shallow learning technique (Logistic Regression, SVM or, a single fully-connected layer etc.) for extracting the unique information from individual modalities as:

1. FMs has linear time complexity and it models both first and second-order factorized interactions from feature vector which enhances its expressive capabilities over other shallow techniques.

2. Real-world datasets often consist of missing values and FMs are capable of dealing with sparsity as they model feature interactions with factorized representations.

Prior to utilizing FMs, the feature vectors from unimodalities are processed via sub-embeddings vectors denoted as f_{FM_a}, f_{FM_v} and f_{FM_l} to extract latent features from the $\boldsymbol{z}_a$ (acoustic), $\boldsymbol{z}_v$ (visual), and $\boldsymbol{z}_l$ (language) respectively. The sub-embeddings network for acoustic and visual modalities is a single feed-forward linear layer. For language modality the sub-embeddings network comprises of LSTM [46] followed by a single feed-forward layer. FMs are then trained independently on f_{FM_v}, f_{FM_a}, f_{FM_l} to obtain y_V, y_A, and y_L as predicted sentiment from their respective modalities. We briefly discuss the details of FMs before presenting the procedure of unique information extraction.

Factorization Machine

FMs were originally proposed for recommendation systems [98]. They are widely utilized for information extraction especially when dealing with extremely sparse feature sets. Given a sparse real valued feature $\boldsymbol{x} \in \mathbb{R}^n$, FMs estimates the target

i.e. $\hat{y}_{FM(\boldsymbol{x})} \in \mathbb{R}$ by modelling all interactions between each pair of features via factorized interaction parameters as below:

$$\hat{y}_{FM(\boldsymbol{x})} = w_0 + \sum_{i=1}^{n} \boldsymbol{w}_i \boldsymbol{x}_i + \sum_{i=1}^{n} \sum_{j=i+1}^{n} \boldsymbol{v}_i^T \boldsymbol{v}_j \; . \; \boldsymbol{x}_i \boldsymbol{x}_j \tag{4.2}$$

where w_0 is the global bias, $\boldsymbol{w} \in \mathbb{R}^n$ models the interaction of the i-th feature to the target. The $\boldsymbol{v}_i^T \boldsymbol{v}_j$ term denotes the factorized interaction, where $\boldsymbol{v}_i \in \mathbb{R}^k$ denotes the latent vector of size k for feature i, and $\hat{y}_{FM(\boldsymbol{x})}$ is the predicted value.

Extracting Acoustic-Specific Unique Information

The latent embeddings denoted as $\boldsymbol{f}_{FM_a} \in \mathbb{R}^{1 \times k_a}$ are obtained from the acoustic features z_a as below:

$$\boldsymbol{f}_{FM_a} = \sigma\Big(\boldsymbol{z}_a \times W_{FM_a} + \boldsymbol{b}_{0_{FMa}}\Big) \tag{4.3}$$

where W_{FM_a} and $\boldsymbol{b}_{0_{FMa}}$ are the sub-embedding network hyper-parameters and σ is the activation function. The unique acoustic information is obtained by utilizing FM in Equation (4.2) on the latent embedding obtained as $y_A = \hat{y}_{FM(\boldsymbol{f_{FMa}})}$.

Extracting Visual-Specific Unique Information

The latent embeddings, $\boldsymbol{f}_{FM_v}$ from z_v are obtained analogous to the acoustic sub-embedding network. The unique visual information is then obtained as $y_V = \hat{y}_{FM(\boldsymbol{f_{FMv}})}$.

$$\boldsymbol{f}_{FM_v} = \sigma\Big(\big(\boldsymbol{z}_v \times W_{FM_v}\big) + \boldsymbol{b}_{0_{FMv}}\Big) \tag{4.4}$$

Extracting Language-Specific Unique Information

The latent embeddings denoted as $\boldsymbol{f}_{FM_l} \in \mathbb{R}^{1 \times k_a}$ are obtained from the language features z_l as below:

$$\boldsymbol{f}_{FM_l} = \sigma\Big(LSTM\big(\boldsymbol{z}_l\big) \times W_{FM_l} + \boldsymbol{b}_{0_{FMl}}\Big) \tag{4.5}$$

where W_{FM_l} and $\boldsymbol{b}_{0_{FM_l}}$ are the sub-embeddings networks hyper-parameters. The unique language specific information is then obtained as $y_L = \hat{y}_{FM(\boldsymbol{f_{FMl}})}$.

4.4.2 Common Network

To obtain the common information from multi-mode representations, we propose a deep convolution-tensor network. In this regard, we first obtain joint representation as tensors from modalities by performing outer product on their latent embeddings. These tensors are naturally multi-dimensional where each element of the tensor represents the interaction strength between the elements of the fusion-modalities. Therefore we applied convolution-kernels on these tensors as they are non-linear feature extractors and generalize better than feed-forward layers [57]. Utilizing convolutions on the joint representations alleviates the need of factorization in **DeepCU** and also makes it highly scalable.

Analogous to the unique network the unimodal representations are processed via sub-embeddings networks to obtain latent embeddings. Then the outer product is utilized to capture joint representations as tensors from these embeddings. Convolution kernels of appropriate dimensions are then applied to the tensors for feature extraction. To reduce the impact of missing values in our common network we obtain multiple sets of combined representation as below:

- T_{AV} bi-modal representation from acoustic & visual.
- T_{AL} bi-modal representation from acoustic & language.
- T_{VL} bi-modal representation from visual & language.
- T_{AVL} tri-modal representation from acoustic, visual, & language.

The motivation to obtain multiple sets of tensor representation is that if assuming any one of the modalities (for example acoustic) has missing values. Then, the tensorial representations obtained with the latent embeddings of this modality (i.e. T_{AV}, T_{AL} and T_{AVL}) are affected but not the other tensor representations

(i.e. T_{VL}). Moreover this information loss is further subsided by the information obtained by the corresponding unique network. Again, the latent embeddings for each tensor pair in the common network are unshared which enables **DeepCU** to obtain complementary information within each tensorial representation.

Extracting Combined Information from the Bi-Modal Interactions of Acoustic and Visual Modalities

The latent embeddings for the acoustic ($f_{AV} \in \mathbb{R}^{1\times k_a}$) and visual features ($f_{VA} \in \mathbb{R}^{1\times k_v}$) are obtained as below:

$$\begin{aligned} f_{AV} &= \sigma\Big(\boldsymbol{z}_a \times W_{av} + b_{av}\Big) \\ f_{VA} &= \sigma\Big(\boldsymbol{z}_v \times W_{va} + \boldsymbol{b}_{va}\Big) \\ T_{AV} &= f_{AV} \otimes f_{VA} \end{aligned} \tag{4.6}$$

where $[W_{av}, b_{av}]$ and $[W_{va}, b_{va}]$ represent the sub-embeddings networks hyper-parameters. $T_{AV} \in \mathbb{R}^{k_v \times k_l}$ represents the bi-modal representation obtained by taking outer product of the latent embeddings. Convolution filters are then applied to capture the non-linear interactions in T_{AV} as:

$$\mathcal{G}_{AV} = \sigma\Big(Conv\big(T_{AV}\big)\Big) \tag{4.7}$$

where $\mathcal{G}_{AV}$ represents the output from convolution layer which is then processed through fully-connected layer as:

$$\boldsymbol{h}_{AV} = \sigma\Big(\hat{\boldsymbol{g}}_{AV} \times W_{AV} + \boldsymbol{b}_{AV}\Big) \tag{4.8}$$

Finally, the hidden representation $\boldsymbol{h}_{AV}$ is processed through feed-forward layer to obtain the final predicted value y_{AV} as:

$$y_{AV} = \big(\boldsymbol{h}_{AV} \times \boldsymbol{w}_{AV}\big) + b_{0_{AV}} \tag{4.9}$$

Extracting Combined Information from the Bi-Modal Interactions of Visual and Language Modalities

The latent embeddings for the visual ($f_{VL} \in \mathbb{R}^{1\times k_v}$) and language features ($f_{LV} \in \mathbb{R}^{1\times k_l}$) are obtained as shown below:

$$\begin{aligned} f_{VL} &= \sigma\Big(\boldsymbol{z}_v \times W_{vl} + b_{vl}\Big) \\ f_{LV} &= \sigma\Big(LSTM\big(\boldsymbol{z}_l\big) \times W_{lv} + \boldsymbol{b}_{lv}\Big) \end{aligned} \tag{4.10}$$

$T_{VL} \in \mathbb{R}^{k_v \times k_l} c$ is then obtained by taking the outer product of the latent embeddings representing their bi-modal interactions. Analogous to Equations (4.7) to (4.9) the bi-modal interactions are processed to obtain y_{VL} as the predicted output in Equation (4.11).

$$\begin{aligned} \mathcal{G}_{VL} &= \sigma\Big(Conv\big(T_{VL}\big)\Big) \\ \boldsymbol{h}_{VL} &= \sigma\Big(\hat{\boldsymbol{g}}_{VL} \times W_{VL} + \boldsymbol{b}_{VL}\Big) \\ y_{VL} &= \big(\boldsymbol{h}_{VL} \times \boldsymbol{w}_{VL}\big) + b_{0_{VL}} \end{aligned} \tag{4.11}$$

Extracting Combined Information from the Bi-Modal Interactions of Acoustic and Language Modalities

Analogous to the above y_{AL} is obtained as the predicted output from bi-modal acoustics and visual interactions in Equation (4.12).

$$\begin{aligned} f_{AL} &= \sigma\Big(\boldsymbol{z}_a \times W_{al} + b_{al}\Big) \\ f_{LA} &= \sigma\Big(LSTM\big(\boldsymbol{z}_l\big) \times W_{la} + \boldsymbol{b}_{la}\Big) \\ \mathcal{G}_{AL} &= \sigma\Big(Conv\big(T_{AL}\big)\Big) \\ \boldsymbol{h}_{AL} &= \sigma\Big(\hat{\boldsymbol{g}}_{AL} \times W_{AL} + \boldsymbol{b}_{AL}\Big) \\ y_{AL} &= \big(\boldsymbol{h}_{AL} \times \boldsymbol{w}_{AL}\big) + b_{0_{AL}} \end{aligned} \tag{4.12}$$

Extracting Combined Information from the Tri-Modal Interactions of Acoustic, Visual and Language Modalities

The tri-modal interactions are obtained by taking outer product between latent embeddings of acoustic, visual and language; i.e. $T_{AVL} = \big(f_{AVL} \otimes f_{VLA} \otimes f_{LAV}\big) \in \mathbb{R}^{k_a \times k_v \times k_l}$. Convolution filters and fully connected layers are then applied on T_{AVL} to obtain the predicted values y_{AVL} as below.

$$\begin{aligned} \mathfrak{G}_{AVL} &= \sigma\Big(Conv\big(T_{AVL}\big)\Big) \\ \boldsymbol{h}_{AVL} &= \sigma\Big(\hat{\boldsymbol{g}}_{AVL} \times W_{AVL} + \boldsymbol{b}_{AVL}\Big) \\ y_{AVL} &= \big(\boldsymbol{h}_{AVL} \times \boldsymbol{w}_{AVL}\big) + b_{0_{avl}} \end{aligned} \tag{4.13}$$

4.4.3 Fusion Layer

The scalar outputs from the common and the unique sub-networks are integrated by applying $\hat{y} = h^T Z$, where the vector Z is obtained by concatenating the predicted scalar outputs from the unique and common sub-networks as $Z = \Big[y_A, y_V, y_L, y_{AL}, y_{VL}, y_{AV}, y_{AVL}\Big]$, and $h = \Big[\hat{h}_A, \hat{h}_V, \hat{h}_L, \hat{h}_{AL}, \hat{h}_{VL}, \hat{h}_{AV}, \hat{h}_{AVL}\Big]$ is a vector of appropriate dimension consisting of fusion weights. For simplicity, all the weights in h can be set to one i.e. $h = J_{1,7}$ and are not optimized while training. We refer this model as static fusion denoted as **DeepCU**$_{SF}$. Otherwise, the weights in h can be randomly initialized (simply a fully connected layer with number of neurons equal to seven) and optimized via the loss on the target function and the model is referred as dynamic fusion denoted as **DeepCU**$_{DF}$.

Our proposed **DeepCU** can be applied to a variety of tasks such as for classification, ranking etc. However, for this work, we estimate the parameters of **DeepCU** via minimizing the mean square error (MSE) loss in Equation (4.14).

$$L = \frac{1}{n} \sum_{\forall x \in \chi} (\hat{y}(x) - y(x))^2 \tag{4.14}$$

where χ denotes the set of multimodal training data instances, $y(x)$ denotes the target of instance x, and $\hat{y}(x)$ denotes the prediction obtained from **DeepCU**.

4.4.4 Complexity Analysis

Theoretically, the paramount computational cost in **DeepCU** is feature extraction from the multimodal tensor which is $\mathcal{O}(N \times K \times S^2 \times M^2)$ as described in [41]) where N and K are the number of input and output feature maps respectively and; S represents the spatial size of the filter and M represents the spatial size of the output feature map. If we fix the dimensionality of the latent space for each modality as 32 (as in LMF), then the number of parameters in **DeepCU** are 1.06e6 whereas the number of parameters in LMF and TFN is equal to 1.1e6 and 12.5e6 respectively.

4.5 Experimental Settings

4.5.1 Dataset

We perform experiments on the CMU-MOSI [132] and POM [86] datasets consisting of YouTube videos for movie reviews. The CMU-MOSI dataset consists of movie reviews videos from 93 distinct speakers. Each video consists of multiple opinion segments with a total of 2199 segments in the whole dataset, annotated with the sentiment in the range $[-3,\ 3]$. The POM dataset consists of 903 movie review videos where each video is annotated 16 sentiments of the speaker. To evaluate the generalization capability of models, the training, validation, and testing splits of the dataset are speaker independent.

Features

We accessed the language, visual, and acoustic features provided by the authors [132] at their official publicly available repository*. The modality specific features are provided after word alignment using P2FA [128] aligning them at the word granularity.

Language

Pre-trained 300-dimensional Glove word embeddings [89] were utilized to encode each sequence of transcribed word into a sequence of word vectors.

Visual

The library Facet† is used to extract visual features for each frame (sampled at 30Hz). Extracted features consists of 20 facial action units, 68 facial landmarks, head pose estimates, gaze tracking and HOG features [137].

Acoustic

COVAREP acoustic framework [28] is utilized to extract features including 12 MFCCs, pitch, glottal source, peak, slope, voiced/unvoiced segmentation, and maxima dispersion quotient.

4.5.2 Baselines

We extensively evaluate the performance of both neural based and non-neural based fusion schemes for multimodal sentiment analysis. We trained our **DeepCU** as well as other benchmarks for regression objective but C-MKL is trained for binary classification due to the objective function utilized in [91]. To calculate the binary

*https://github.com/A2Zadeh/CMU-MultimodalSDK, SDK Version 1.0.1

†https://imotions.com/

and multi-class accuracies we followed the protocol in [74] and maps the predicted output (and the target values) to integer values.

Early Fusion, Non-Neural Approaches

SVM (Support Vector Machines), this baseline is trained on the concatenated multimodal features for regression task.

RF (Random Forest), this baseline is also trained on the concatenated multimodal features for regression tasks.

Joint Representation, Neural Approaches

DNN_{JR} [90, 83] is a deep neural network trained by concatenating features from sub-embeddings networks for regression.

SVM-MD [132] is the SVM classifier trained with joint representation obtained from the hidden layers of DNN_{JR}.

RF-MD similar to $SVM\text{-}MD$, is trained random forest on joint representations obtained from the hidden layers of DNN_{JR}.

C-MKL [91] is a deep Convolution-MKL network which trains multiple kernel learning [115] on features obtained from DNN_{JR}.

ELM [93] is an extreme learning machine trained on multimodal dictionary of joint representation obtained from DNN_{JR}.

DeepFM [38] is a wide and deep classifier leveraging the power of both FMs and deep networks for multimodal data fusion.

The latent space for learning FMs and deep networks were obtained using sub-

embeddings networks as in our **DeepCU**.

State of the art deep networks for Multimodal data fusion

$Hybrid_{DS}$ (Hybrid-DeepShallow) [124] is a hybrid architecture integrating deep and shallow networks with Random Forest for multimodal data fusion (described in Section 4.3.3). The information from acoustic and visual modalities are extracted using deep networks while language representations are modelled using SVM.

TFN (Tensor Fusion Networks) [129] is multimodal deep tensor network trained on the joint representation obtained via tensor product of multi-mode representations in Equation (4.1). The official code is provided at author's GitHub repository[‡].

LMF (Low-rank Multimodal Fusion) [74] is current state of the art multimodal deep fusion classifier (with no contextual information) trained on low-rank factorized modes of the joint representation. The official code is provided at author's GitHub repository[§].

4.5.3 Parameter Settings in DeepCU

We train our model by minimizing the MSE loss with RMS optimizer with learning rate equal to 6×10^{-3} and batch-size of 64. To avoid over-fitting we applied dropout [109] in our model and tune the dropout probability from [0.1, 0.9] with a step size of 0.05. The optimal dimensions of latent spaced within each sub-embeddings network was searched in [5,10,15,20,25,30], while the number of convolution filters was searched in [1, 2, 3, 4, 5]. We also varied the size of convolution filter between 3 and 5. Moreover, to reduce covariance shift and improve performance we applied batch normalization [50] to all hidden layers of **DeepCU**. For

[‡]https://github.com/Justin1904/TensorFusionNetworks

[§]https://github.com/Justin1904/Low-rank-Multimodal-Fusion

acoustic and visual modalities the sub-embeddings network is a single feed-forward layer, while for language we used LSTM [46] (basic uni-directional LSTM cell) with 128 units.

4.5.4 Evaluation Metrics

We evaluate the performance of the baselines and **DeepCU** for regression, binary and multi-class classification problems. For regression, we report Mean Absolute Error (MAE) and Pearson's Correlation (Correlation). For binary classification, we report accuracy and F1 score, where as for multi-class classification we only report accuracy. For all metrics higher value is better except for MAE. Similar to [129, 74] we employed early stopping strategy, where we terminated training **DeepCU** and all baselines if the MAE on validation-set did not improved in 5 consecutive epochs.

4.5.5 Results and Explainability Analysis

The key contribution of this work is utilization of both unique and common information for multimodal data fusion. We performed experiments to study the significance of our proposed fusion scheme under the following research questions:

Q1: Does the integration of common and unique latent information actually beneficial or their integration deteriorates the performance of **DeepCU** over individual sub-networks?

To evaluate this, we studied whether fusing the common and unique information is actually beneficial or their integration deteriorates the performance of **DeepCU** over individual sub-networks. To achieve so, we evaluate the performance of common network on all the hyper-parameter settings as explained in Section 4.5.3. While the unique network were evaluated by varying the size of latent dimensions and dropout probabilities and optimizers. We also applied both the dynamic and static fusion schemes to the common and unique networks. We present the MAE of the

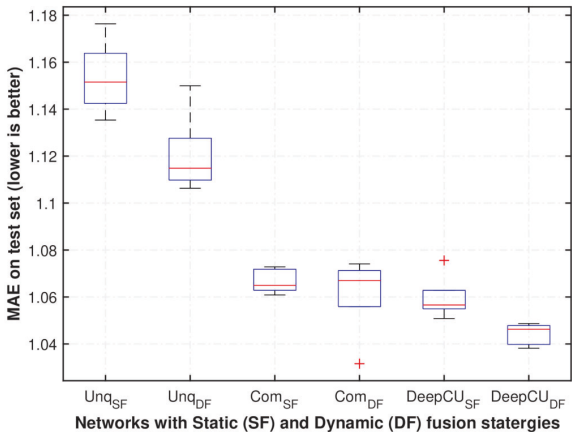


Figure 4.3 : Performance comparison of DeepCU vs common (Com) and unique (Unq) networks on the CMU-MOSI dataset.

optimized networks with box-plot in Figure 4.3.

It is clearly visible that integrating both the common and the unique information improves the performance of proposed **DeepCU**. The common network exploits the information from both bi-modal and tri-modal interactions by applying deep-convolution operations which drastically reduces its MAE compared to unique networks. Besides, the plot suggests that for all the networks the dynamic fusion performs slightly better than static fusion. However the network with dynamic fusion layer required more epochs for convergence.

Besides, the integration of common and unique information further achieves reduction in MAE and is visible in the box plots for both the fusion schemes in **DeepCU**. Moreover, **DeepCU** with dynamic fusion scheme achieves the lowest MAE and confirms that integration of common and unique information is actually beneficial for multimodal data fusion.

Q2: Are convolutions able to efficiently capture the information from non-linear interactions exhibiting in the multi-mode representation? Moreover, how does the

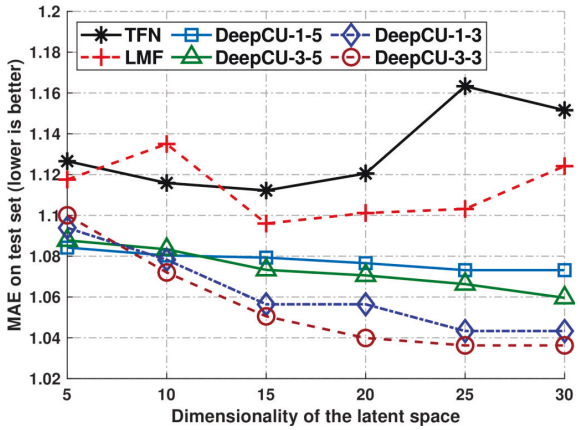


Figure 4.4 : Performance of DeepCU, TFN, and LMF by varying hyperparameters on the CMU-MOSI dataset. The legend DeepCU-x-y represents, x = number of convolution filters and y = filter size.

hyper-parameters affect the performance of **DeepCU**?

We now present a detailed study on how hyper-parameters affects the performance of **DeepCU**. In this regard, we plot the mean MAE obtained by varying hyper-parameters in Figure 4.4(b). The x-axis in plot represents the dimensionality of latent-embeddings and the curves represents combinations on a) the number of convolution filters, b) filter-size, and c) fusion scheme. We also plot the performances of TFN and LMF obtained on the same latent dimensions.

A clear trend can be seen in all the curves reflecting performance of **DeepCU**, where the MAE tends to decrease with increase in the latent dimensions. This is because the lower dimensions tensor is equal to the size of convolution kernel and hence the performance of **DeepCU** is not significantly better than TFN and LMF. However, the marginal improvement can be attributed to the unshared latent space and the unique information. Besides, the performance gradually improves with the increase in the latent dimensions which supports the learning requirement of convolution kernels.

MOSI Dataset	Regression		Binary		7-class
	MAE (lower is better)	Correlation	Accuracy	F1	Accuracy
RF	$1.4095 \pm 1.09\times10^{-4}$	$0.2041 \pm 3.29\times10^{-4}$	$53.98 \pm 6.57\times10^{-1}$	52.75 ± 1.48	18.27 ± 1.41
SVM	$1.4259 \pm 1.43\times10^{-5}$	$0.1288 \pm 3.36 \times10^{-4}$	47.74 ± 5.78	36.59 ± 4.37	$13.98 \pm 4.12\times10^{-1}$
DNN_{JR} [90]	$1.1801 \pm 2.31\times10^{-4}$	$0.4973 \pm 2.41\times10^{-4}$	$68.59 \pm 2.27\times10^{-1}$	$68.67 \pm 2.27\times10^{-1}$	25.48 ± 3.75
RF-MD	$1.1993 \pm 1.63\times10^{-4}$	$0.4636 \pm 2.41\times10^{-4}$	$66.11 \pm 5.81\times10^{-1}$	$66.16 \pm 6.02\times10^{-1}$	$26.03 \pm 3.60\times10^{-1}$
SVM-MD [132]	$1.2749 \pm 2.97\times10^{-4}$	$0.4950 \pm 1.71\times10^{-4}$	$67.60 \pm 2.59\times10^{-1}$	$67.68 \pm 2.64\times10^{-1}$	$17.49 \pm 1.00\times10^{-1}$
C-MKL [91]	—	—	$66.85 \pm 4.65\times10^{-1}$	$68.30 \pm 6.43\times10^{-1}$	—
ELM [93]	$1.1786 \pm 2.28 \times10^{-4}$	$0.4935 \pm 1.22 \times10^{-4}$	69.70 ± 1.08	71.61 ± 1.66	24.42 ± 1.68
DeepFM [38]	$1.1038 \pm 1.81 \times10^{-5}$	$0.5227 \pm 1.73 \times10^{-4}$	$69.14 \pm 7.64 \times10^{-1}$	$69.10 \pm 7.3 \times10^{-1}$	$28.90 \pm 4.54 \times10^{-1}$
$Hybrid_{DS}$ [124]	$1.4919 \pm 2.56 \times10^{-2}$	$0.1350 \pm 9.29 \times10^{-3}$	50.92 ± 1.41	48.85 ± 2.32	$16.44 \pm 2.04\times10^{-1}$
TFN (SOTA 1) [129]	$1.1111 \pm 3.03 \times10^{-4}$	$0.5341 \pm 1.02 \times10^{-4}$	$69.59 \pm 7.06 \times 10^{-1}$	$68.48 \pm 7.93 \times 10^{-1}$	31.98 ± 1.13
LMF (SOTA 2) [74]	$1.0960 \pm 2.11 \times10^{-4}$	$0.5555 \pm 3.28 \times10^{-5}$	$70.25 \pm 2.05 \times 10^{-1}$	$70.31 \pm 1.98 \times 10^{-1}$	$30.76 \pm 3.39 \times 10^{-1}$
DeepCU$_{SF}$ (static fusion)	$1.0595 \pm 7.08 \times10^{-5}$	$0.5536 \pm 7.66 \times10^{-5}$	$71.49 \pm 2.00 \times 10^{-1}$	$71.42 \pm 1.98 \times 10^{-1}$	$33.54 \pm 6.39 \times 10^{-1}$
DeepCU$_{DF}$ (dynamic fusion)	$\mathbf{1.0442 \pm 1.71} \times10^{-5}$	$\mathbf{0.5609 \pm 1.05} \times10^{-5}$	$\mathbf{73.54 \pm 1.10} \times 10^{-1}$	$\mathbf{73.52 \pm 1.14} \times 10^{-1}$	$\mathbf{34.04 \pm 3.61} \times 10^{-1}$

Table 4.2 : Performance comparison of DeepCU vs other fusion techniques on CMU-MOSI dataset. The mean and variance for each baseline and DeepCU are obtained by executing them for five times. This superiority of DeepCU is specifically visible in the case of 7-class classification.

Another trend can be noticed in the performance curves of **DeepCU** where convolutions of filter-size 3 performs slightly better that filter-size 5. This may be due to the increase in overlapping regions between segments which might be better for applying convolution on multi-mode representations.

Q3: Does **DeepCU** provide a better mulit-modal fusion technique compared to state of the art such as TFN and LMF?

We compare the performance of multiple SOTA (and other baselines) and **DeepCU** on the CMU-MOSI and POM datasets for this requirement and the results are reported in Sections 4.5.5 and 4.5.5.

On CMU-MOSI dataset we improve the state of the art by **4.68**% for regression and **2.25**% for correlation and on multi-class the accuracy improvement is **9.63**%. On POM dataset we improve the correlation by **23.10**% and for regression the

POM Dataset	MAE	Correlation	Multi-Class Accuracy
TFN (SOTA 1)	1.0481 ± 0.0030	0.0866 ± 0.023	28.62 ± 0.127
LMF (SOTA 2)	0.8739 ± 0.0051	0.2311 ± 0.024	33.61 ± 0.314
DeepCU$_{DF}$	**0.8568 ± 0.0045**	**0.2845 ± 0.009**	**34.77 ± 0.493**

Table 4.3 : Performance comparison on the POM dataset.

improvement is **2**% compared to state of the art.

The above results confirms our hypothesis on the advantages of **DeepCU**: a) utilizing both the common and unique latent information obtained using unshared-embeddings; b) the use of convolutions to capture utmost expressiveness offered by multi-mode representation; and c) the use of factorized representations in unique networks to reduce the impact of missing values present in the individual modalities.

As expected the dynamic fusion schemes performs better than the static fusion scheme in **DeepCU**. Conceptually, this is because the weights in the static fusion layer were not optimal and optimizing these weights via back-propagation allows the proposed **DeepCU**$_{DF}$ network to obtain better mixing weights for integrating common and unique information.

4.5.6 Case Study with Missing Values from the Acoustic Modality in the CMU-MOSI Dataset

As a qualitative analysis on the performance of the fusion schemes, we perform an investigative study of TFN, LMF, and **DeepCU** when facing missing values in the feature sets. In this regard, we selected two examples with highest percentage of missing values from the actual dataset in the acoustic modality and reported their predicted sentiment obtained from each of the fusion schemes in Table 4.4.

Missing Values in the Acoustic Modality	Ground-truth of the Sentiment	TFN	LMF	DeepCU
63.51 %	0.0	0.5118	-0.3387	**-0.0154**
21.62 %	-1.0	-1.3475	-1.4417	**-1.1209**

Table 4.4 : Affect of missing values on DeepCU$_{DF}$, TFN, and LMF. These feature vectors are taken from the actual CMU-MOSI dataset.

In the first example, the absolute error with the prediction from TFN is 0.5118, from LMF is 0.3387; and from **DeepCU** is 0.0154. The predicted sentiment value from **DeepCU** achieves the lowest error when the corresponding feature set contains a large fraction of missing values. In the second example, the absolute error with the prediction from TFN is 0.3475, from LMF is 0.4417; and from **DeepCU** is 0.1209. Again the predicted sentiment value from **DeepCU** achieves the lowest error when the corresponding feature set contains moderate fraction of missing values.

These examples confirms the effectiveness of utilizing both common and unique information for multimodal data fusion. Moreover, they also exhibit the importance of handling missing values with real-world datasets, as their proper consideration might boost the performance of multimodal systems.

4.6 Summary

In this chapter, we have introduced **DeepCU** which utilizes both common and unique latent information for sentiment analysis on multimodal data. The **DeepCU** consolidates two sub-networks a) deep convolution-tensor networks for obtaining common information from multi-model data; and b) unique subnetwork to obtain information offered by the individual modalities. Both the sub-networks are integrated via a fusion layer, and the parameters are optimized by back-propagation on the target loss function. The **DeepCU** outperformed state of the art approaches as it leverages the expressiveness of all-types of information by enforcing the two sub-networks to learn complimentary information in the embeddings layer. Comprehensive experiments demonstrate the effectiveness of our proposed **DeepCU** for multimodal data fusion.

Chapter 5

Towards Effective Data Augmentations via Unbiased GAN Utilization

5.1 Introduction

In today's era of big data, artificially intelligent products and services are increasingly deployed in our daily lives. The manifestation of these machine learning (ML) models range from medical diagnosis to personalized-user scores such as bank loan approvals, and image recognition [127, 44, 12]. The complexity of decision rules for state of the art deep neural networks has increased exponentially, which in turn has resulted in high overall decision accuracy on many benchmark datasets. However, at the same time, the predictions of these highly accurate models often have reflected systematic biases for identifiable minority subgroups [59, 82]. As inherent problems within the datasets such as dataset-bias (which is usually overlooked) affect the decisions of the ML models leading to false predictions for the minority subgroups [8]. The consequences of such false predictions can be catastrophic, for example, the Uber self-driving car's accident* and the racial biases in Google searches†. Hence, relying on accuracy as the sole criteria for social deployment of these models is undesirable [58].

Although the benchmark datasets for training these models are built with an intent to capture the real-world representations, at the same time, strong built-in biases are rhetorically evidenced in these datasets [111, 112]. While corrective

*https://www.bbc.com/news/technology-44243118

†http://www.bbc.com/news/technology-21322183

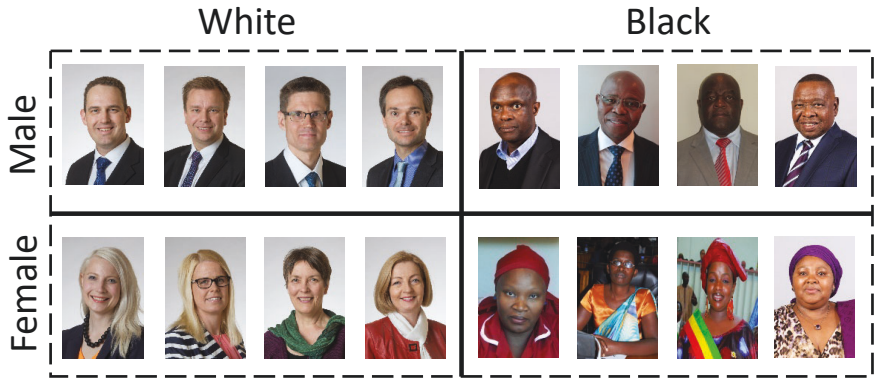


Figure 5.1 : Images from PPB dataset [11].

measures for well-known dataset biases such as the capture-bias[‡] and label-bias[§] (also known as category bias) are studied in the literature [94, 21]. Unfortunately, multiple unknown biases still remain hidden in the dataset and cause poor generalization performance of machine learning models.

In this regard, a promising empirical evaluation for the above scenario is presented in [11], where the authors created a test-bed named PPB for gender recognition. The PPB dataset was created with demographic parity based on Fitzpatrick skin types and example images shown in Fig. 5.1. The authors demonstrated that despite low error rates achieved by the commercial classifiers on benchmark gender recognition dataset as claimed by their manufactures. The misclassification rate of these classifier's on PPB test-set is biased towards darker skin individuals. Moreover, the bias in these classifiers is remarkably significant for darker skin females.

An over-simplified solution to alleviate this phenomenon is to remove the culprit data instances. However, the identification of such data instances is a challenge, and more importantly, the performance of models trained on unbiased dataset might

[‡]Related to the device utilized while capturing the data instances; it is also related to collectors view or preferences for the real world.

[§]Arises when the visual categories are poorly defined, like similar images may be annotated with different names by annotators.

deteriorate [81]. This will lead to a rollback of the previous biased model, which contradicts the objective of removing biases from the datasets.

Another solution can be auditing the model with a validation set to discover possible fractures in the model and then retraining them [21]. However, retraining a model might not be cost-effective or bring any benefit for the model creators. Moreover, the validation set utilized by the auditor is also prone to dataset biases and might have been created from a completely different distribution. Hence this setting is also not optimal.

Motivated by the issues mentioned above, in this chapter, we address the issue of bias management in the datasets by developing a data provisioning mechanism which we call as **Data Augmentation Pursuit** (**DAP**). Contrary to previous works, where sophisticated machine learning models are devised to mitigate the dataset-biases while learning ML models [55, 81], *we are interested in how we can use the available data to augment these datasets with synthetic instances, resulting in lesser bias learned by the ML models.*

To achieve this objective, we utilize generative adversarial networks (GANs) [35] to generate synthetic examples for augmenting the existing datasets. However, we argue that blindly augmenting the datasets with synthetic examples generated by GANs does not guarantee a reduction in bias learned by the machine learning models [123]. Rather the bias in the augmented dataset might increase. Therefore, a principled approach is required to augment these datasets. In this regard, we devise **Data Augmentations Pursuit** (**DAP**). The **DAP** objectives are to ensure that the retained synthetic examples do not increase the biases while augmenting the datasets and employ a customized iterative filtering scheme for the same. The ML models thus trained on the augmented datasets obtains better performance than the original model and exhibits a decrease in the biases learned from the dataset.

5.2 Our Contributions

Our contributions in this are summarized as below:

1. We propose **Data Augmentation Pursuit** (**DAP**) for augmenting dataset with synthetic examples. The **DAP** regulates the fraction of sample inputs to GAN and controls the synthetic examples selection for dataset augmentaion. ML models trained with the obtained augmented using **DAP** exhibits least model and achieves significantly better classification performance.

2. We propose a filtering strategy for sieving synthetic examples generated by GAN. Our filtering strategy ensures the reduction in semantic gap between real and synthetically generated data instances.

3. We perform extensive experimentation on CIFAR-10 dataset by utilizing various GAN's frameworks for data augmentation and empirically demonstrate that proper attention is required while augmenting datasets.

The rest of the chapter is organized in the following sections: Literature review is presented in Section 5.3, followed by Section 5.3.2 on preliminaries of GANs. Our proposed **DAP** is described in Section 5.4 and finally experiments, results, and conclusions are discussed in Section 5.5, Section 5.6, and Section 5.7 respectively.

5.3 Realted Work

Data augmentation has played a crucial role in object and image recognition tasks. To improve recognition accuracy using CNN, several high performing models have applied extensive data augmentation to their training datasets [65, 110]. Conventionally, for generating synthetic examples, trivial image transformation techniques like random rotation, cropping, contrast normalization, etc. were utilized.

However, not all synthetic examples help in improving the classifier's learning algorithm and selecting good examples is critically important [88]. Moreover, for large datasets, the number of possible data augmentations are exhaustive, and the number of parameters in CNN is exponential. Hence selecting good synthetic examples is almost intractable. Therefore, we require a clever procedure to obtain synthetic data instances which increase the value of the datasets and the classifiers inexpensively. We focus our literature review on the work which augments the training data by adding "virtual samples" following a systematic procedure and not blindly applying basic image transformations.

Paulin et. al.[88] proposed a novel approach for creating augmented data sets by greedily selecting set of image transformations. Their proposed technique, i.e. "Image Transformation Pursuit" (ITP) iteratively and greedily selects a set of optimal transformations which maximizes the classifier's performance. While performing prediction with the trained classifier, the optimal transformations obtained by ITP are first applied to each test instances, and then these transformed instances are classified. Similarly, in [81], the authors proposed sophisticated data augmentations which exist in the real-world scenarios but might not exists in the training dataset. Performance of classifiers trained with their proposed augmentations generalizes better on cross-datasets.

Similarly, Sato et. al. in [101] proposed an online data augmentation procedure called APAC (Augmented PAttern Classification), which applies random deformations to the data samples in an online fashion. Here the classifier is only trained with multiple deformed samples of the training data instances. The expected loss from multiple deformed instances of the same true data is utilized for training the classifier. Similar to ITP, the testing data instance undergoes the same deformation process while performing predictions. However, both ITP and APAC requires heavy computational resources. Hence, their proposed technique is not applicable when

deeper networks with an exponential number of parameters require augmentations.

Conversely, Khosla et. al. [55]s proposed a discriminative framework that explicitly defines bias associated with each dataset and, attempts to approximate weights for the generalization. Their model applies the max-margin principle to perform better on cross datasets by taking into account the label of the originating datasets for the data instances. Their model can be considered as a sophisticated domain adaptation technique which simultaneously trains a classifier on multiple datasets.

5.3.1 GAN utilization in dataset augmentation and their limitations

A natural applicability for synthetic image generated using GANs is dataset augmentation, and some recent techniques have indeed utilized the GANs for dataset augmentation for training deep convolution neural networks. In [105], the authors proposed refinement of synthetic images by processing them with a refiner trained on unlabeled real data called *SimGAN*. The refiner adds realism to the synthetic images such that the synthetic images look similar to the real image but preserves the annotated information of the generator. Classifiers trained with these refined images improves the state of the art in gaze estimation. Similarly, in [61], the authors proposed refinement of synthetic images by conditioning on the image quality and achieved improvement in presentation attacks in biometric applications. However, each of the above-devised mechanism has to apply domain-specific knowledge to increase the quality of generated images before their utilization, and hence, their applicability is limited.

Another recent work in [3], namely Data Augmentation GAN (*DAGAN*), is proposed to learn a set of data augmentation for a target domain. The *DAGAN* learns to obtain a data instance from a source domain and augments the target domain by generating a within-class sample in it. Although the generated image by *DAGAN* is accepted as a different sample of the target domain, the image generation

process still faces the dataset bias issue, as explained in the next subsection.

5.3.2 Dataset Bias and GANs

The generative adversarial networks (**GAN**s) first introduced in [35] are usually composed of two deep neural networks. The first network is called the discriminator (**D**), while the second network is called the generator (**G**). The generator network aims to generate realistic images starting from random prior (z) resembling true images of the dataset. In other words, if p_x is the distribution over true data then $\mathbf{G}(z)$ learns the distribution $p_g \sim p_x$. On the other hand, **D** aims at learning the discrimination between the distributions p_x and p_g, where $\mathbf{D}$(input) represents the probability $(p_x|input)$ and $\mathbf{G}(z)$ represents the output from **G** having noise (z) as its input. Both the networks compete against each other in minimax two-player game objective and optimize their parameters with alternative update rules as defined in Equation (5.1) and Equation (5.2) respectively, where m denotes the batch size.

$$\Delta_{\theta_d} \frac{1}{m} \sum_{i=1}^{m} [logD(x^i) + log(1 - D(G(z^i)))] \tag{5.1}$$

$$\Delta_{\theta_g} \frac{1}{m} \sum_{i=1}^{m} [log(1 - D(G(z^i)))] \tag{5.2}$$

The gradient-based update of the generators **G**'s parameters i.e., (θ_g) in Equation (5.2) are dependent on the outcome of the discriminator **D** and its parameters i.e. (θ_d) are updated prior updating parameters of **G**. Due to this update strategy, the bias leaned by the discriminator network eventually gets inherited by the learning mechanism of the generator network.

In other words, the discriminator, which is a deep neural network, can easily fall prey to inevitable dataset biases [94, 21]. Consequently, these biases do not just affect the discriminator's decision but also affect the learning mechanism of the generator network. The same phenomenon leads to mode collapse [4] while

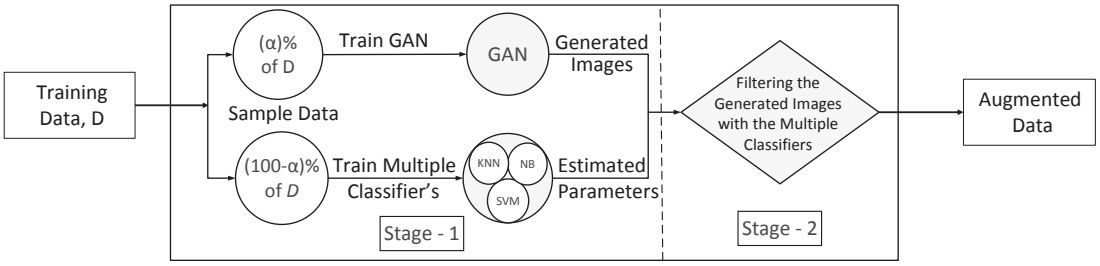


Figure 5.2 : Procedure of Generating Sieved Synthetic Data

training GANs and are currently an active topic of research. Therefore, the resultant augmented dataset will eventually contain these biases inherently affecting any classifiers' learning mechanism.

Moreover, despite the recent advancements in GAN, synthetic images generated by them on datasets with high variabilities like CIFAR or ImageNet are of low quality [37, 123]. Improving the quality of the images generated by GANs is currently an active research topic, but this work does not focus on improving the learning framework of GANs. Instead, this work focuses on how one selects a subset of images to train GANs such that, the generated synthetic images can be utilized to augment the training dataset.

5.4 Data Augmentation Pursuit

In a nutshell, our work is similar to ITP, as both targets selection of synthetic examples for augmenting the datasets. But our work differs in two ways: 1) we focus on harnessing the gains from available synthetic images generated by GANs, whereas ITP first selects the transformations to augment the dataset and then generates synthetic examples accordingly. 2) In ITP, both training and testing data were augmented, while we only augment the training dataset and do not alter the testing dataset.

As explained in Sec. 5.3.2, blindly augmenting datasets with synthetic examples

can increase the bias in the augmented datasets. Therefore, we design a two-stage filtering technique to control the training data instances utilized to train GANs and sieve unbiased synthetic examples generated by the generator. Our filtering technique is based on the ensemble classifier learning, which outperforms a single classifier by creating diversity in the ensemble [66]. As a result, this leads to a reduction in bias on the final prediction from an ensemble classifier [8]. Hence, synthetic images selected using **DAP** does not adversely affect the learning system of classifier's when trained on them.

Moreover, due to filtering of synthetic images with an ensemble classifier; synthetic instances which closely resembles true data distribution receives consensus on the prediction from classifiers' in the ensemble. As a result, the semantic gap between true data and synthetic data is reduced and, augmenting datasets with these filtered images results in reducing the variance learned by the alternating models which reduce the effects dataset biases in the learning mechanism of ML models. Our two-stage filtering technique is shown in Fig. 5.2.

5.4.1 Stage-1.

Randomly sample $\alpha\%$ of data instances from the true dataset (denoted as D) to train conditional GAN. A conditional GAN is simply a GAN framework conditioned with certain priors. This conditioning helps in generating synthetic examples by selecting the priors in the generator. Once the GAN is trained, we generate an adequate number of synthetic examples denoted as I by conditioning the generator with data labels as priors. The utilization of conditional GAN generates synthetic examples with known ground truth. Simultaneously, we utilize $(100 - \alpha)\%$ of the remaining true dataset to train our ensemble classifiers (naive Bayes, SVM, and KNN).

The motivation behind splitting the dataset D in '$\alpha\%$' and '$(100 - \alpha)\%$' while

Algorithm 4 *Data Augmentation Pursuit*

Input: Training Data D, Train labels $T_L \in \mathbb{Z}_2^M$, splitting percentage α

$GAN_\Theta \leftarrow$ Train GAN on α% of D

$I \leftarrow$ generate synthetic examples by trained GAN conditioned on T_L

$E_\theta \leftarrow$ Train SVM, naive Bayes, and KNN classifiers on the remaining $(100 - \alpha)$% of D

$Pred_L \leftarrow \theta(I)$, predict the labels for synthetic examples using ensemble classifiers

$Index \leftarrow$ select the indices from $Pred_L$ where ensemble classifiers has consensus (majority vote) on the prediction and the synthetic example is correctly classified

$[D_{Aug},\ D_{Lab}] \leftarrow I[Index],\ T_L[Index]$ retain synthetic examples filtered from above

Output: Augmented Data $\left[D_{Aug}\right]$ and Augmented Label $\left[D_{Lab}\right]$

training GAN and ensemble classifier is to ensure that the biases learned by the two sub-processes are dissimilar. Later in **Stage-2** when filtering synthetic images generated by GANs utilizing ensemble classifier, the biases of the two sub-processes will work against each other, resulting in the elimination of synthetic examples which are misclassified by the ensemble classifier.

5.4.2 Stage-2.

Utilize the pre-trained ensemble classifier from **Stage-1** to classify the synthetic images generated by the GANs. The synthetic images which are correctly classified and achieving a consensus from the ensemble classifier are retained. Since the bias learned by the ensemble classifier and the GAN are complementary due to the random split of training data between them. The complementary biases act against each other while filtering synthetic images generated by GAN with ensemble classier. Hence, this strategy cancels the bias learned by the two mechanisms guaranteeing that augmenting dataset with these retained synthetic images will reduce the dataset bias and eventually, the model bias. The whole procedure of augmenting datasets with *Data Augmentation Pursuit* is described in Alg. 4.

5.5 Experiments

Datasets and GAN Frameworks We utilized publicly available implementation of DCGAN[¶] [95] and IWGAN[‖] [37] architectures on CIFAR-10 dataset [65]. The dataset consists of natural RGB images of size 32×32 distributed among 10 categories. Since we require labelled synthetic data generation the implementation for DCGAN was modified by conditioning both the discriminator and the generator on input labels.

However, the current state of GANs are not able to generate images which can span the whole manifold of the training data, i.e., can be utilized for training ML models [104, 123]. We downscale our experiments to binary categories as this reduces the search space required by the generator drastically and recognizable synthetic images are generated. Furthermore, the bootstrap sampling parameter 'α' Alg. 4 is initialized with a value equal to 10% of the true data and is incremented with 10% on each iteration in Alg. 4.

5.5.1 Experimental Setup and Performance Metric

In our experiments, we compare the performance of the CNN[**] classifier on four datasets 1) original CIFAR-10 dataset 'Org'; 2) dataset augmented blindly with synthetic examples generated using DCGAN[95] 'DCGAN'; 3) dataset augmented blindly with synthetic examples generated using IWGAN [37]; and 4) dataset augmented by applying two stage filtering strategy of **DAP**.

For evaluating the performance of the classifier trained on the above datasets, we utilized three performance metrics: a) classification accuracy, b) bias and c) variance. We performed 3-$fold$ cross-validation on multiple binary categories and reported the

[¶]https://github.com/kvfrans/generative-adversial

[‖]https://github.com/igul222/improved_wgan_training

[**]https://github.com/soumith/DeepLearningFrameworks

Algorithm 5 Calculation of Bias Variance and Accuracy

Input: Training Data D_{train}, Training Label L_{train}, Testing Data D_{test}, Testing Label L_{test}, Augmented Data D_{Aug}, Augmented Label L_{Aug}, Cross-folds = k, $\alpha \in [0, 10, 20, ..., 90, 100]$

$(D_1, L_1), (D_2, L_2), ..., (D_k, L_k) \leftarrow CV(D_{train}, L_{train})$ ▷ Create K cross folds of the training data and training labels

$(D_{A_1}, L_{A_1}), (D_{A_2}, L_{A_2}), ..., (D_{A_k}, L_{A_k}) \leftarrow CV(D_{train}, L_{train})$ ▷ Create K cross folds of the augmented data and augmented label

$[Pred_{Label},\ Accuracy] \leftarrow [\,],\ [\,]$

for iter = 1 to k **do**

  if isequal$(\alpha, 0)$ **then**

    $[train,\ label] \leftarrow D_{iter}, L_{iter}$ ▷ Use true training data and labels

  else

    $train \leftarrow [D_{iter}, D_{A_{iter}}]$ ▷ add synthetic examples to training data

    $label \leftarrow [L_{iter}, L_{A_{iter}}]$ ▷ add synthetic labels to training labels

  $Model_{\Theta} \leftarrow \mathbf{CNN}(train, label)$ ▷ Train model parameters on the training data

  $[Pred_{Label}, Accuracy] \leftarrow \mathbf{CNN}(test, \Theta_{CNN})$ ▷ predict labels and accuracy of testing examples using CNN and append them to the List

$Accuracy \leftarrow mean(Accuracy)$ ▷ calculate mean of k-fold accuracies

$Bias \leftarrow bias^2(L_{test}, Pred_{Label})$ ▷ calculate bias using Eq. 5.3

$Variance \leftarrow variance(Pred_{Label})$ ▷ calculate bias using Eq. 5.4

Output: Bias, Variance, Accuracy

mean accuracy, whereas the bias and variance inherited learning mechanism of any classifier can be obtained by bias-variance decomposition technique for zero-one loss functions [60] and are mathematically calculated as below:

$$bias_x^2 \equiv \frac{1}{2} \sum_{y \in Y} \big[P(\, Y_F = y|x) - P(\, Y_H = y|x) \big]^2 \tag{5.3}$$

$$variance_x \equiv \frac{1}{2} \left(1 - \sum_{y \in Y} [P(\, Y_H = y|x)^2 \right) \tag{5.4}$$

where, Y_F represents the ground truth of data instance x represented as a probability distribution (one hot vector), and Y_H represents the probability distribution for the predictions made by the classifier for the data instance x.

5.5.2 Feature Extraction for ensemble classifier

We utilized K-means triangle features [22] for training ensemble classifier in stage-1 Fig. 5.2 of **DAP**. The process begins with extracting random sub-patches from the input data neglecting its labels, denoted as $\mathbf{X} \in \mathbb{R}^{M \times N}$, where M is the total number of sub-patches and each sub-patch $x_i \in \mathbb{R}^N$ and $i \in [1, M]$. The vectors in $\mathbf{X}$ are then normalized by subtracting the mean and dividing them by the standard deviation of its elements, followed by a whitening procedure. After pre-processing, the K-means clustering technique is applied to learn 'k' centroids $c^{(k)}$. Finally for each $x_i \in X$, K-means triangle features are extracted [22]. Briefly, K-means triangle features are a form of non-linear mapping where each feature f_k is encoded with the following rule.

$$f_k(x) = max\big\{0, \mu(z) - z_k\big\} \tag{5.5}$$

where $z_k = \|x - c^{(k)}\|_2$ and $\mu(z)$ is the mean of the elements of z. This mapping assigns '0' for any feature f_k where the distance from $c^{(k)} > \ \mu(z)$.

5.6 Results and Discussions

In this section, we study the performance of CNN and SVM classifiers with various degrees of dataset augmentation. We divide the discussion into two subsections, wherein the first subsection, we study the performance of the CNN classifier trained on the four datasets described in Section 5.5.1. Our hypothesis of measuring the model bias consists of three performance metrics, namely the bias, variance, and the accuracy of the classifier. In the second subsection, we study the effect on the classifier's performance by regularizing the level of data augmentation, i.e., by varying the hyper-parameter α in proposed **DAP**.

5.6.1 How does data-augmentation affect the performance of classifier?

In order to evaluate the above research question, we study the performance of the CNN and SVM classifiers when trained on following datasets

1. original dataset, i.e., data without any augmentation.
2. dataset augmented blindly with synthetic examples generated using DCGAN or IWGAN.
3. dataset augmented with synthetic examples obtained with our proposed **DAP**.

We evaluate the bias, the variance, and the accuracy of the CNN classifier on 20 randomly selected pairs from the CIFAR-10 dataset. The results are presented as four columns in Tabel. 5.1, where the first three columns reflect the performance of the CNN classifier on the original dataset (column 1), the dataset augmented blindly with synthetic examples from DCGAN (column 2); and the dataset augmented blindly with synthetic examples from IWGAN (column 3). The last column (i.e., **DAP** column) reflects the classifier performance evaluated on the augmented dataset obtained using **DAP**. The values under the **DAP** column are chosen according to the optimal value of the α parameter, i.e., the value where the maximum reduction in the bias of the classifier is attained.

Similarly, we evaluate the bias, the variance, and the accuracy of SVM classifier on 20 randomly selected pairs from the CIFAR-10 dataset, and the results are shown in Table. 5.2. Again, the values under the **DAP** column are chosen according to the optimal value of , α, i.e., the value where the maximum reduction in the bias of the classifier is attained.

Besides, to test the significance of our developed approach, we use paired t-test to test the null hypothesis that *the difference between the two distributions in the pair comes from the same normal distribution.* Where each pair consists of the values

Table 5.1 : Performance comparisons using CNN classifier on baselines datasets and augmented dataset obtained using **DAP**. The *p-values* obtained using *t-tests* on pairs 'Baseline vs **DAP**' are tabulated in the last column. Note that we follow the scientific notation $**$ where we use 1Ex to present 1×10^x. Please note that, for the bias and the variance lower value is better whereas for accuracy higher is better.

Categories	**Accuracy**				**Bias**				**Variance**			
	Org	DCGAN	IWGAN	**DAP**	Org	DCGAN	IWGAN	**DAP**	Org	DCGAN	IWGAN	**DAP**
Frog - Truck	.962	.965	.961	**.976**	.028	.026	.029	**.016**	.010	.009	.010	**.006**
Frog - Ship	.965	.973	.963	**.976**	.027	.020	.026	**.017**	.008	.006	.010	**.005**
Cat - Truck	.942	.946	.933	**.953**	.047	.042	.051	**.036**	.011	.012	.016	**.010**
Bird - Truck	.954	.960	.946	**.963**	.034	.030	.041	**.027**	.012	.010	.013	**.009**
Dog - Ship	.954	.962	.956	**.964**	.034	.030	.034	**.027**	.012	**.008**	.009	.009
Mobile - Cat	.953	.958	.952	**.960**	.038	.032	.035	**.031**	.009	.010	.013	**.007**
Dog - Truck	.958	.962	.945	**.965**	.031	.030	.041	**.026**	.011	.008	.014	**.007**
Frog - Horse	.951	.954	.951	**.961**	.034	.034	.036	**.028**	.015	.012	.014	**.011**
Mobile - Dog	.967	.971	.968	**.973**	.025	.023	.024	**.021**	.008	**.006**	.008	**.006**
Horse - Truck	.953	.952	.942	**.959**	.034	.036	.044	**.030**	.013	.012	.014	**.009**
Deer - Dog	.852	.870	.859	**.871**	.110	.099	.106	**.096**	.038	**.031**	.035	.032
Plane - Truck	.921	.925	.913	**.928**	.060	.056	.065	**.053**	.019	.019	.021	**.017**
Deer - Frog	.896	.905	.902	**.908**	.075	.069	.069	**.066**	.029	.027	.030	**.025**
Dog - Frog	.906	.913	.921	**.919**	.067	.065	.057	**.060**	.026	.022	.022	**.021**
Mobile - Bird	.964	.966	.957	**.968**	.026	.025	.029	**.023**	.011	.010	.014	**.009**
Mobile - Horse	.977	.982	.972	**.980**	.015	.013	.021	**.013**	.008	**.006**	.007	**.006**
Plane - Ship	.899	.908	.903	**.910**	.074	.069	.072	**.066**	.027	.023	.025	**.024**
Mobile - Frog	.971	.969	.960	**.973**	.022	.022	.029	**.019**	.008	.009	.011	**.007**
Mobile - Deer	.975	.974	.967	**.979**	.018	.018	.023	**.016**	.007	.007	.010	**.005**
Dog - Horse	.858	**.875**	.866	.874	.105	**.095**	.102	**.095**	.037	**.030**	.033	**.030**
p-values	1E−8	3E−5	1E−8	-	3E−8	2E−6	1E−7	-	6E−8	6E−5	7E−9	-

obtained through the baselines (i.e., the values under the column *Org, DCGAN, IWGAN*) against values obtained through **DAP** as shown in Table. 5.1. The last row of Table. 5.1 and Table. 5.2 reflects the $p - value$ of the t-statistics obtained at 5% level of significance. These low $p - values$ reject the null hypothesis, and the improvements achieved using proposed **DAP** are statistically significant.

Table 5.2 : Performance comparison using SVM classifier on baselines datasets and augmented dataset obtained using **DAP**. The *p-values* obtained using *t-tests* on pairs 'Baseline vs **DAP**' are tabulated in the last column. Note that we follow the scientific notation ∗∗ where we use 1Ex to present 1×10^x. Please note that, for the bias and the variance lower value is better whereas for accuracy higher is better.

Categories	**Accuracy**				**Bias**				**Variance**			
	Org	DCGAN	IWGAN	**DAP**	Org	DCGAN	IWGAN	**DAP**	Org	DCGAN	IWGAN	**DAP**
Plane - Cat	.934	.936	.909	**.945**	.047	.042	.059	**.037**	.018	.021	.030	**.016**
Mobile - Frog	.975	.979	.966	**.980**	.018	.016	.021	**.014**	.006	.004	.011	**.004**
Frog - Ship	.979	.977	.962	**.983**	.015	.015	.024	**.012**	.005	.006	.012	**.004**
Mobile - Bird	.969	.972	.954	**.973**	.020	.019	.027	**.017**	.010	.009	.017	**.008**
Horse - Truck	.960	.960	.941	**.969**	.026	.027	.039	**.022**	.012	.012	.018	**.007**
Plane - Mobile	.941	.943	.933	**.950**	.041	.039	.042	**.035**	.017	.017	.023	**.015**
Mobile- Deer	.978	.980	.961	**.982**	.014	.012	.024	**.012**	.007	.007	.0144	**.005**
Mobile - Horse	.973	.974	.960	**.979**	.016	.017	.025	**.014**	.009	.008	.013	**.006**
Plane - Truck	.929	.931	.910	**.936**	.050	.046	.058	**.043**	.020	.021	.031	**.018**
Bird - Ship	.952	.949	.937	**.955**	.032	.034	.040	**.028**	.015	.016	.022	**.013**
Mobile - Ship	.941	.941	.934	**.945**	.040	.037	.044	**.035**	.017	.020	.021	**.017**
Plane - Horse	.951	.948	.934	**.957**	.033	.036	.041	**.030**	.014	.015	.024	**.010**
Ship - Truck	.937	.941	.926	**.945**	.043	.042	.051	**.039**	.018	.016	.022	**.013**
Cat - Truck	.953	.954	.941	**.958**	.032	.030	.037	**.028**	.014	.015	.020	**.011**
Dog - Truck	.963	**.969**	.952	.967	.025	**.021**	.030	.023	.011	.009	.017	**.008**
Plane - Bird	.895	.892	.879	**.904**	.070	.072	.079	**.064**	.034	.035	.040	**.029**
Plane - Frog	.969	.968	.944	**.972**	.022	.023	.035	**.020**	.008	.008	.020	**.006**
Bird - Deer	.856	.853	.847	**.864**	.099	.097	.102	**.091**	.044	.048	.050	**.042**
Frog - Horse	.958	.959	.950	**.961**	.028	.028	.033	**.026**	.013	.012	.015	**.009**
p-values	1E−7	9E−6	9E−12	-	1E−5	1E−4	1E−10	-	5E−8	2E−7	1E−12	-

**https://en.wikipedia.org/wiki/Scientific_notation

It is clearly visible from the evaluations that the improvement in classification performance is achieved via the reduction in bias within the models trained on augmented datasets obtained using our proposed augmentation service. Moreover, this validates our claims that one must not blindly augment the datasets with synthetic examples generated by GANs to achieve higher recognition performance. Instead, proper attention must be given to the dataset bias, which is the root cause of algorithmic bias in ML models responsible for their catastrophic failures.

5.6.2 How does the percentage of input data affect the quality of data-augmentation?

In order to evaluate, how does our bootstrap parameter α controls the bias in training dataset, we plot the performance of classifiers by varying α (data split percentage in Stage-1 of **DAP**) between 10% to 90% of the training data.

The accuracy, bias, and variance of the CNN classifier with various amount of data-split during Stage-1 of **DAP** is shown in Fig. 5.3, Fig. 5.4, Fig. 5.5 respectively. Note that, the y-axis in the plots are scaled for visualization.

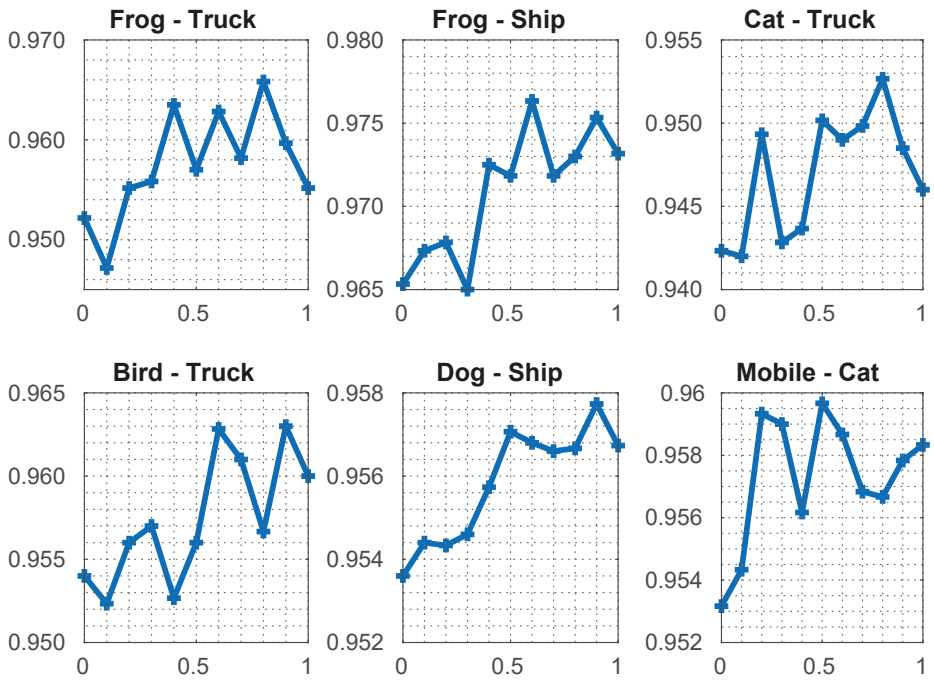


Figure 5.3 : Accuracy-plot of top 6 pairs from Table 5.1. x-axis in subplots represents the values of α used in experiments, where $x = 0, 1$ corresponds to *Org, DCGAN*. The y-axis represents the mean accuracy obtained after 3-fold *crossvalidation*.

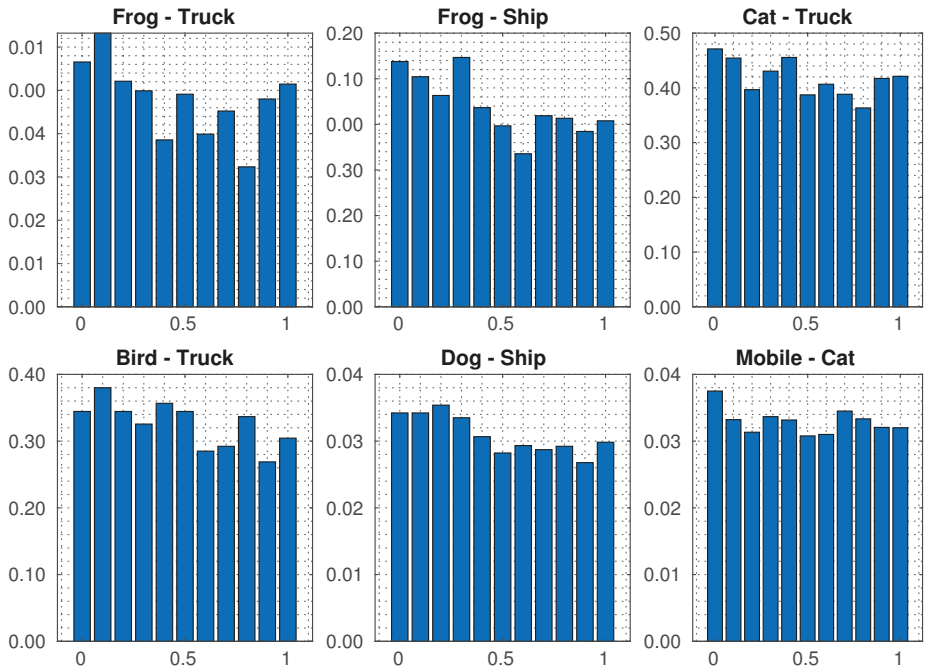


Figure 5.4 : Bias-plot of top 6 pairs from Table 5.1. The x-axis represents the values of α used in our experiments and y-axis represents the bias of the classifier.

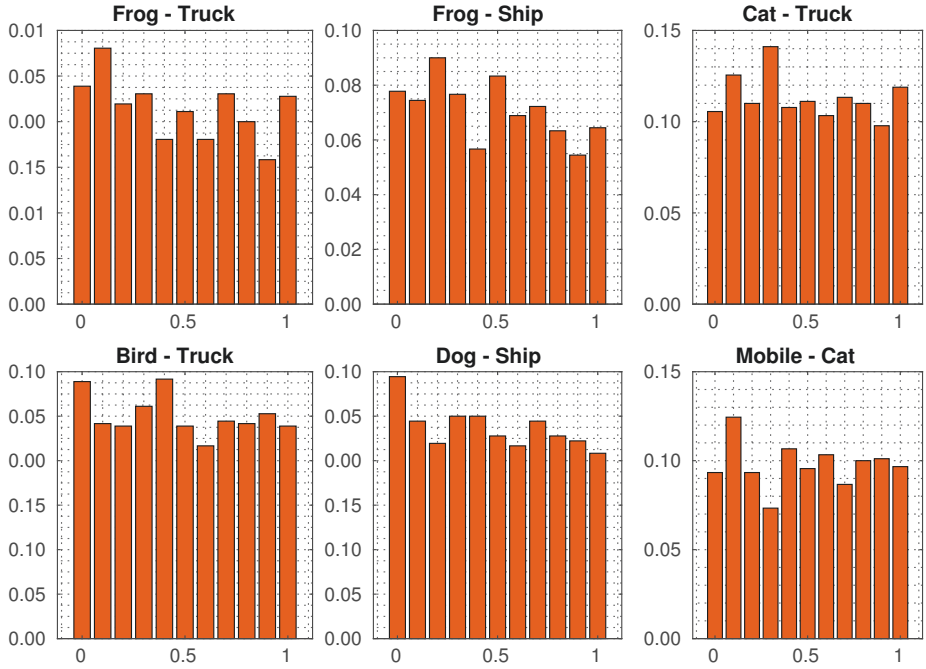


Figure 5.5 : Variance-plot of top 6 pairs from Table 5.1. The x-axis represents the values of α used in our experiments and y-axis represents the bias of the classifier.

Similarly, the accuracy, bias, and variance of the SVM classifier with various amount of data-split is shown in Fig. 5.6, Fig. 5.7, and Fig.5.8 respectively. Note that, the y-axis in the plots are scaled for visualization.

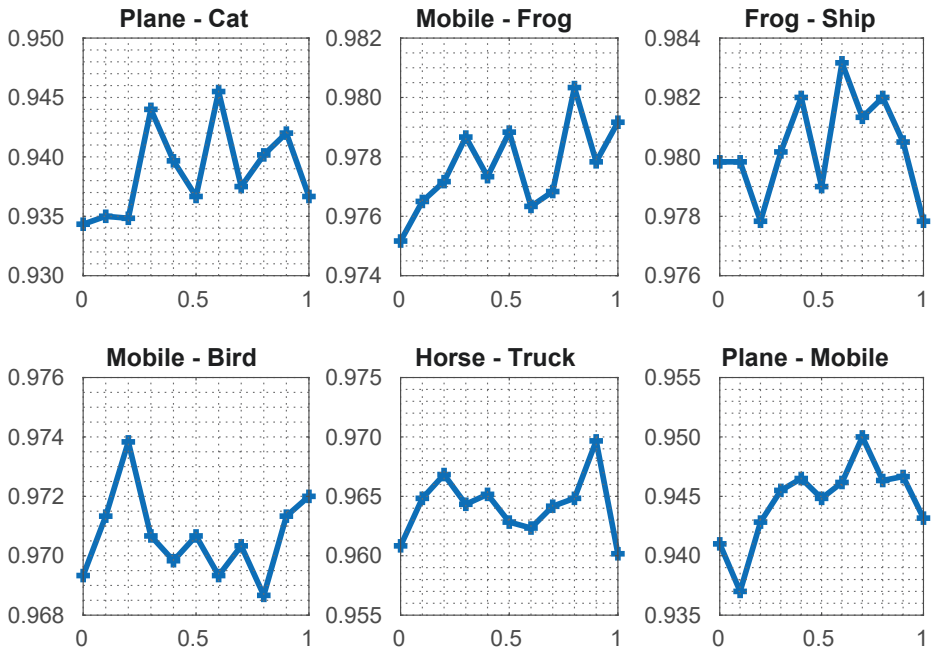


Figure 5.6 : Accuracy-plot of top 6 pairs from Table 5.1. x-axis in subplots represents the values of α used in experiments, where $x = 0, 1$ corresponds to *Org, DCGAN*. The y-axis represents the mean accuracy obtained after 3-fold *crossvalidation*.

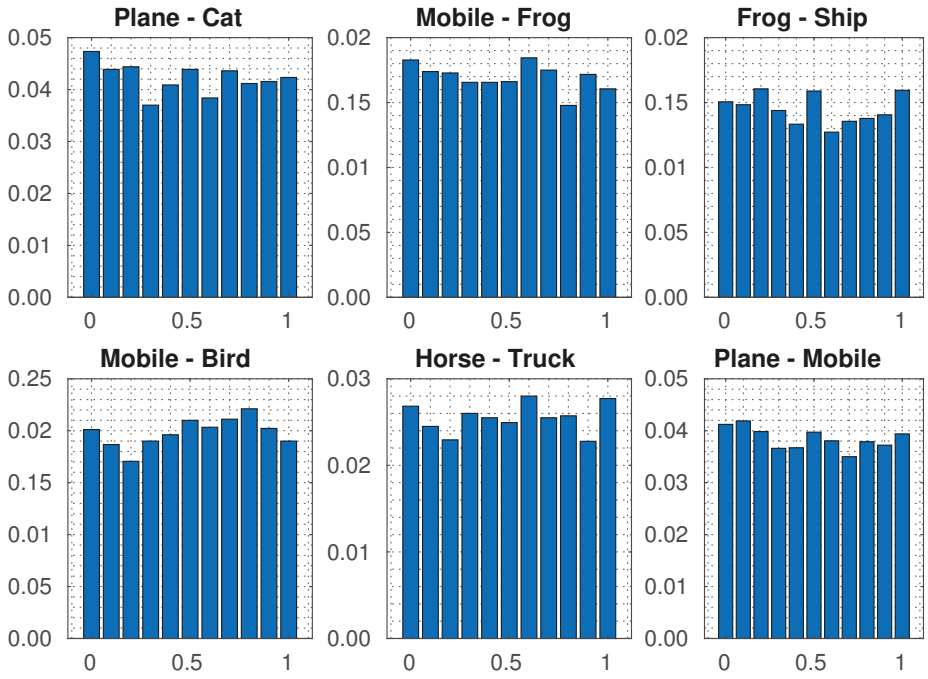


Figure 5.7 : Bias-plot of top 6 pairs from Table 5.2. The x-axis represents the values of α used in our experiments and y-axis represents the bias of the classifier.

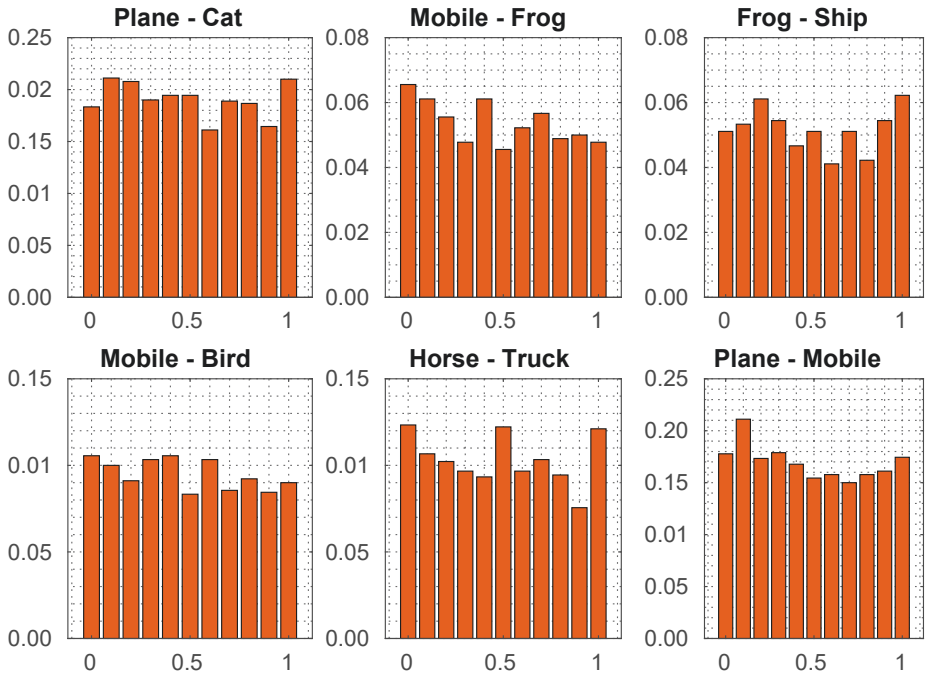


Figure 5.8 : Variance-plot of top 6 pairs from Table 5.2. The x-axis represents the values of α used in our experiments and y-axis represents the bias of the classifier.

While the performance of the classifiers in these plots fluctuates, their performances are mostly better than the baseline, i.e., a) no data-augmentation ($x-axis = 0$) and b) augmenting dataset blindly ($x - axis = 1$). The reason for performance drop at certain α (for example, Plane-Mobile accuracy plot in Fig. 5.6) can be due to the mode-collapsing in GAN [4].

Also, the highest peak in accuracy for CNN classifier in Fig. 5.3 can be followed with corresponding reductions in bias Fig. 5.4 and the variance Fig. 5.5. Analogously, the same correspondence can be drawn with the accuracy, bias, and variance plots of SVM classifier in Fig .5.6, Fig. 5.7, and Fig .5.8 respectively.

5.7 Summary

In this chapter, we presented a formal analysis of bias and variance associated with the learning system of GANs and their effects on the bias of the learning systems of classifiers. The proposed data augmentation strategy **DAP** is empirically shown to alleviate the effects of dataset bias induced in the ML model. ML models trained on augmented datasets obtained with **DAP** shows a reduction in their bias and achieves significantly better classification performance on multiple binary categories of CIFAR -10. Besides, the results measuring the bias and variance on classifier's learning system advocate the need for effective bias management while augmenting datasets with synthetic images generated using GANs.

Chapter 6

Conclusion and Future Work

In this chapter, a summary of the thesis is presented and future research directions are highlighted that could be extended from the research contained in this thesis.

6.1 Conclusions of the Thesis

The performance of any machine learning system is heavily dependent on the feature representation utilised for the task at hand. In the era of big data, artificially intelligent systems strive to utilise information from multiple cues, to obtain a complete knowledge of the phenomenon of interest. However, to utilise available information, one needs to address these two challenging questions, 1) why we need to combine information from multiple cues, and 2) how to perform their fusion. This thesis addressed these questions, by designing feature extraction techniques to obtain common and unique information from multiview, multimodal, and multisource data and proposed new fusion techniques for their utmost utilisation. Below is a summary of the contributions of this research.

- Chapter 3 described the development of a lightweight deep network for image classification. The proposed **Attn-HybridNet** extracts both the unique information from the amalgamated view and the common information from the minutiae view of the data. The two kinds of information are then combined by using attention fusion to obtain final feature representation of an image. The proposed **Attn-HybridNet** achieves competitive performance with significantly less computational resources compared to other similar deep neural

networks.

- In Chapter 4 I introduced a deep neural network called **DeepCU**, which utilises both common and unique latent information for sentiment analysis with multimodal data. The **DeepCU** consolidates two sub-networks, a) deep convolution-tensor network for obtaining common information from multimodal data, and b) a unique sub-network to obtain information offered by the individual modalities. Both sub-networks are integrated via a fusion layer, and their parameters are optimised by back-propagation on the target loss function. The **DeepCU** outperformed state of the art approaches as it leveraged the expressiveness of all-types of information by enforcing the two sub-networks to learn complimentary information in the embedding layer.

- In Chapter 5, I addressed the problem of dataset bias, by designing a data provisioning mechanism called **DAP**. The DAP comprised of two-stage mechanism to control and filter the training data instances 1) utilized to train GANs and 2) sieve unbiased synthetic examples generated by the GAN. The **DAP**'s sieving technique induces diversity in the augmented datasets, thus reduces the variance associated with the unique distribution of the majority subclasses. Hence, synthetic images selected using **DAP** do not adversely affect the learning system of classifier's and significantly improve their classification performance.

6.2 Recommendations for Future Work

This thesis has presented novel algorithms to improve supervised classification with the utilisation of both common and unique information extracted from the data (both heterogeneous and homogeneous). These developments suggest promising research directions that can be extended from this work.

- **Non-linear filter design in Attn-HybridNet.**

  The weights for convolution layers in the parsimonious deep neural network studied in this research are extracted using *principal components* and *LoMOI*, both of which are linear. However, multiple non-linearities in the data, such as image occlusion, alignment, etc. demand the design of a sophisticated filter for accommodating these non-linearities.

- **Develop better fusion technique for DeepCU.**

  In this research I have extracted the unimodal, bimodal, and trimodal information from multimodal data and have combined them with either static or dynamic fusion schemes in the fusion layer. Both of these fusion schemes are elementary as they do not consider the quality of the data source or the latent information while combining them. Providing disparate importance such as an attention weighting mechanism to the unimodal, bimodal, and trimodal latent might prove beneficial and improve the quality of sentiment prediction.

- **Data Augmentation Pursuit in multiclass settings.**

  I have presented a data provisioning mechanism in binary-class settings for dataset augmentation. Extending this two-stage filtering procedure for the multiclass setting is non-trivial and substancially challenging, as the GANs are still not powerful to generate diverse synthetic images with (unknown) distribution equal to the training datasets. A future research direction for this

work can be the design of an incremental model that updates the target classes for augmenting training datasets with synthetic examples.

Bibliography

[1] 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 *OSDI*, vol. 16, 2016, pp. 265–283.

[2] A. Amini, A. Soleimany, W. Schwarting, S. Bhatia, and D. Rus, "Uncovering and mitigating algorithmic bias through learned latent structure," 2019.

[3] A. Antoniou, A. Storkey, and H. Edwards, "Augmenting image classifiers using data augmentation generative adversarial networks," in *International Conference on Artificial Neural Networks*. Springer, 2018, pp. 594–603.

[4] M. Arjovsky, S. Chintala, and L. Bottou, "Wasserstein generative adversarial networks," in *International Conference on Machine Learning*, 2017, pp. 214–223.

[5] T. Baltrušaitis, C. Ahuja, and L.-P. Morency, "Multimodal machine learning: A survey and taxonomy," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2018.

[6] Y. Bengio, A. Courville, and P. Vincent, "Representation learning: a review and new perspectives," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 35, no. 8, pp. 1798–1828, 2013.

[7] D. Berthelot, T. Schumm, and L. Metz, "Began: Boundary equilibrium generative adversarial networks," *arXiv preprint arXiv:1703.10717*, 2017.

[8] D. Brain and G. I. Webb, "The need for low bias algorithms in classification learning from large data sets," in *PKDD*, vol. 2. Springer, 2002, pp. 62–73.

[9] R. E. Broadhurst, "Statistical estimation of histogram variation for texture classification," in *Proc. Intl. Workshop on Texture Analysis and Synthesis*, 2005, pp. 25–30.

[10] J. Bruna and S. Mallat, "Invariant scattering convolution networks," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 35, no. 8, pp. 1872–1886, 2013.

[11] J. Buolamwini and T. Gebru, "Gender shades: Intersectional accuracy disparities in commercial gender classification," in *Conference on Fairness, Accountability and Transparency*, 2018.

[12] D. Cao, L. Nie, X. He, X. Wei, S. Zhu, and T.-S. Chua, "Embedding factorization models for jointly recommending items and user generated lists," in *Proceedings of the 40th International ACM SIGIR Conference on Research and Development in Information Retrieval*. ACM, 2017, pp. 585–594.

[13] T.-H. Chan, K. Jia, S. Gao, J. Lu, Z. Zeng, and Y. Ma, "Pcanet: a simple deep learning baseline for image classification?" *IEEE Transactions on Image Processing*, vol. 24, no. 12, pp. 5017–5032, 2015.

[14] W. Chen, J. Wilson, S. Tyree, K. Weinberger, and Y. Chen, "Compressing neural networks with the hashing trick," in *International Conference on Machine Learning*, 2015, pp. 2285–2294.

[15] X. Chen, Y. Duan, R. Houthooft, J. Schulman, I. Sutskever, and P. Abbeel, "Infogan: Interpretable representation learning by information maximizing generative adversarial nets," in *Advances in Neural Information Processing Systems*, 2016, pp. 2172–2180.

[16] J. Cheng, J. Wu, C. Leng, Y. Wang, and Q. Hu, "Quantized cnn: a unified approach to accelerate and compress convolutional networks," *IEEE transactions on neural networks and learning systems*, no. 99, pp. 1–14, 2017.

[17] J.-T. Chien and Y.-T. Bao, "Tensor-factorized neural networks," *IEEE transactions on neural networks and learning systems*, vol. 29, no. 5, pp. 1998–2011, 2017.

[18] S. Chintala. (accessed September 3, 2019). [Online]. Available: https://github.com/soumith/DeepLearningFrameworks

[19] J. Choe, S. Park, K. Kim, J. Hyun Park, D. Kim, and H. Shim, "Face generation for low-shot learning using generative adversarial networks," in *Proceedings of the IEEE International Conference on Computer Vision*, 2017, pp. 1940–1948.

[20] A. Cichocki, D. Mandic, L. De Lathauwer, G. Zhou, Q. Zhao, C. Caiafa, and H. A. Phan, "Tensor decompositions for signal processing applications: From two-way to multiway component analysis," *IEEE Signal Processing Magazine*, vol. 32, no. 2, pp. 145–163, 2015.

[21] D. A. Cieslak and N. V. Chawla, "Detecting fractures in classifier performance," in *Seventh IEEE International Conference on Data Mining (ICDM 2007)*. IEEE, 2007, pp. 123–132.

[22] A. Coates, A. Ng, and H. Lee, "An analysis of single-layer networks in unsupervised feature learning," in *Proceedings of the fourteenth international conference on artificial intelligence and statistics*, 2011, pp. 215–223.

[23] A. Coates and A. Y. Ng, "The importance of encoding versus training with sparse coding and vector quantization," in *Proceedings of the 28th international conference on machine learning (ICML-11)*, 2011, pp. 921–928.

[24] F. Cong, Q.-H. Lin, L.-D. Kuang, X.-F. Gong, P. Astikainen, and T. Ristaniemi, "Tensor decomposition of eeg signals: a brief review," *Journal of Neuroscience Methods*, vol. 248, pp. 59–69, 2015.

[25] M. Crosier and L. D. Griffin, "Using basic image features for texture classification," *International Journal of Computer Vision*, vol. 88, no. 3, pp. 447–460, 2010.

[26] L. De Lathauwer, B. De Moor, and J. Vandewalle, "A multilinear singular value decomposition," *SIAM journal on Matrix Analysis and Applications*, vol. 21, no. 4, pp. 1253–1278, 2000.

[27] ——, "On the best rank-1 and rank-(r 1, r 2,..., rn) approximation of higher-order tensors," *SIAM Journal on Matrix Analysis and Applications*, vol. 21, no. 4, pp. 1324–1342, 2000.

[28] G. Degottex, J. Kane, T. Drugman, T. Raitio, and S. Scherer, "Covarep—a collaborative voice analysis repository for speech technologies," in *Acoustics, Speech and Signal Processing (ICASSP), 2014 IEEE International Conference on*. IEEE, 2014, pp. 960–964.

[29] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "ImageNet: A Large-Scale Hierarchical Image Database," in *CVPR09*, 2009.

[30] N.-E. El Faouzi, H. Leung, and A. Kurian, "Data fusion in intelligent transportation systems: Progress and challenges–a survey," *Information Fusion*, vol. 12, no. 1, pp. 4–10, 2011.

[31] R.-E. Fan, K.-W. Chang, C.-J. Hsieh, X.-R. Wang, and C.-J. Lin, "Liblinear: a library for large linear classification," *Journal of Machine Learning Research*, vol. 9, no. Aug, pp. 1871–1874, 2008.

[32] A. Fukui, D. H. Park, D. Yang, A. Rohrbach, T. Darrell, and M. Rohrbach, "Multimodal compact bilinear pooling for visual question answering and visual grounding," in *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, 2016, pp. 457–468.

[33] X. Glorot and Y. Bengio, "Understanding the difficulty of training deep feedforward neural networks," in *Proceedings of the thirteenth international conference on artificial intelligence and statistics*, 2010, pp. 249–256.

[34] L. Gong, B. Haines, and H. Wang, "Clustered model adaption for personalized sentiment analysis," in *Proceedings of the 26th International Conference on World Wide Web*. International World Wide Web Conferences Steering Committee, 2017, pp. 937–946.

[35] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial nets," in *Advances in neural information processing systems*, 2014, pp. 2672–2680.

[36] K. Grauman and T. Darrell, "The pyramid match kernel: discriminative classification with sets of image features," in *Computer Vision, 2005. ICCV 2005. Tenth IEEE International Conference on*, vol. 2. IEEE, 2005, pp. 1458–1465.

[37] I. Gulrajani, F. Ahmed, M. Arjovsky, V. Dumoulin, and A. Courville, "Improved training of wasserstein gans," in *NIPS*, 2017.

[38] H. Guo, R. Tang, Y. Ye, Z. Li, and X. He, "Deepfm: a factorization-machine based neural network for ctr prediction," in *Proceedings of the 26th International Joint Conference on Artificial Intelligence*. AAAI Press, 2017.

[39] S. Han, H. Mao, and W. J. Dally, "Deep compression: compressing deep neural networks with pruning, trained quantization and huffman coding," *ICLR*, 2016.

[40] E. Hayman, B. Caputo, M. Fritz, and J.-O. Eklundh, "On the significance of real-world conditions for material classification," in *European Conference on Computer Vision*. Springer, 2004, pp. 253–266.

[41] K. He and J. Sun, "Convolutional neural networks at constrained time cost," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2015, pp. 5353–5360.

[42] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 770–778.

[43] X. He and T.-S. Chua, "Neural factorization machines for sparse predictive analytics," in *Proceedings of the 40th International ACM SIGIR conference on Research and Development in Information Retrieval*. ACM, 2017, pp. 355–364.

[44] X. He, M. Gao, M.-Y. Kan, Y. Liu, and K. Sugiyama, "Predicting the popularity of web 2.0 items based on user comments," in *Proceedings of the 37th international ACM SIGIR conference on Research & development in information retrieval*. ACM, 2014, pp. 233–242.

[45] F. L. Hitchcock, "The expression of a tensor or a polyadic as a sum of products," *Journal of Mathematics and Physics*, vol. 6, no. 1-4, pp. 164–189, 1927.

[46] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural computation*, vol. 9, no. 8, pp. 1735–1780, 1997.

[47] G. Hu, Y. Hua, Y. Yuan, Z. Zhang, Z. Lu, S. S. Mukherjee, T. M. Hospedales, N. M. Robertson, and Y. Yang, "Attribute-enhanced face recognition with neural tensor fusion networks." in *ICCV*, 2017, pp. 3764–3773.

[48] G. Huang, Z. Liu, L. Van Der Maaten, and K. Q. Weinberger, "Densely connected convolutional networks," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 4700–4708.

[49] J. Huang and C. Yuan, "Fanet: factor analysis neural network," in *International Conference on Neural Information Processing*. Springer, 2015, pp. 172–181.

[50] S. Ioffe and C. Szegedy, "Batch normalization: accelerating deep network training by reducing internal covariate shift," in *Proceedings of the 32nd International Conference on International Conference on Machine Learning-Volume 37*. JMLR. org, 2015.

[51] Y. Jia, C. Huang, and T. Darrell, "Beyond spatial pyramids: Receptive field learning for pooled image features," in *2012 IEEE Conference on Computer Vision and Pattern Recognition*. IEEE, 2012, pp. 3370–3377.

[52] Y. Jia, O. Vinyals, and T. Darrell, "On compact codes for spatially pooled features," in *International Conference on Machine Learning*, 2013, pp. 549–557.

[53] I. Karmanov. (accessed May 20, 2019) Vgg style cnn on cifar10. [Online]. Available: https://github.com/soumith/DeepLearningFrameworks

[54] T. Karras, T. Aila, S. Laine, and J. Lehtinen, "Progressive growing of gans for improved quality, stability, and variation," *ICLR*, 2018.

[55] A. Khosla, T. Zhou, T. Malisiewicz, A. A. Efros, and A. Torralba, "Undoing the damage of dataset bias," in *European Conference on Computer Vision*. Springer, 2012, pp. 158–171.

[56] H. A. Kiers, "Towards a standardized notation and terminology in multiway

analysis," *Journal of Chemometrics: A Journal of the Chemometrics Society*, vol. 14, no. 3, pp. 105–122, 2000.

[57] D. Kim, C. Park, J. Oh, S. Lee, and H. Yu, "Convolutional matrix factorization for document context-aware recommendation," in *Proceedings of the 10th ACM Conference on Recommender Systems*. ACM, 2016, pp. 233–240.

[58] M. P. Kim, A. Ghorbani, and J. Zou, "Multiaccuracy: Black-box post-processing for fairness in classification," 2019.

[59] J. Kleinberg, S. Mullainathan, and M. Raghavan, "Inherent trade-offs in the fair determination of risk scores," *ITCS*, 2017.

[60] R. Kohavi, D. H. Wolpert *et al.*, "Bias plus variance decomposition for zero-one loss functions," in *Machine Learning, Proceedings of the Thirteenth International Conference (ICML*, 1996, pp. 275–283.

[61] N. Kohli, D. Yadav, M. Vatsa, R. Singh, and A. Noore, "Synthetic iris presentation attack using idcgan," in *IJCB*, 2017.

[62] T. G. Kolda and B. W. Bader, "Tensor decompositions and applications," *SIAM review*, vol. 51, no. 3, pp. 455–500, 2009.

[63] J. Kossaifi, A. Khanna, Z. Lipton, T. Furlanello, and A. Anandkumar, "Tensor contraction layers for parsimonious deep nets," in *Computer Vision and Pattern Recognition Workshops (CVPRW), 2017 IEEE Conference on*. IEEE, 2017, pp. 1940–1946.

[64] A. Krizhevsky. (2012 (accessed May 20, 2019)) Cuda convnet. [Online]. Available: https://code.google.com/archive/p/cuda-convnet/

[65] A. Krizhevsky and G. Hinton, "Learning multiple layers of features from tiny images," *Department of Computer Science, University of Toronto*, 2009.

[66] L. I. Kuncheva and C. J. Whitaker, "Measures of diversity in classifier ensembles and their relationship with the ensemble accuracy," *Machine learning*, vol. 51, no. 2, pp. 181–207, 2003.

[67] D. Lahat, T. Adali, and C. Jutten, "Multimodal data fusion: an overview of methods, challenges, and prospects," *Proceedings of the IEEE*, 2015.

[68] H. Larochelle, D. Erhan, A. Courville, J. Bergstra, and Y. Bengio, "An empirical evaluation of deep architectures on problems with many factors of variation," in *Proceedings of the 24th international conference on Machine learning*. ACM, 2007, pp. 473–480.

[69] V. Lebedev, Y. Ganin, M. Rakhuba, I. Oseledets, and V. Lempitsky, "Speeding-up convolutional neural networks using fine-tuned cp-decomposition," in *ICLR*, 2015.

[70] Y. Li, M. Yang, and Z. M. Zhang, "A survey of multi-view representation learning," *IEEE Transactions on Knowledge and Data Engineering*, 2018.

[71] T.-Y. Lin, A. RoyChowdhury, and S. Maji, "Bilinear cnn models for fine-grained visual recognition," in *Proceedings of the IEEE International Conference on Computer Vision*, 2015, pp. 1449–1457.

[72] M.-Y. Liu and O. Tuzel, "Coupled generative adversarial networks," in *Advances in neural information processing systems*, 2016, pp. 469–477.

[73] W. Liu, J. Chan, J. Bailey, C. Leckie, and K. Ramamohanarao, "Mining labelled tensors by discovering both their common and discriminative subspaces," in *Proceedings of the 2013 SIAM International Conference on Data Mining*, 2013.

[74] Z. Liu, Y. Shen, V. B. Lakshminarasimhan, P. P. Liang, A. Zadeh, and L.-P.

Morency, "Efficient low-rank multimodal fusion with modality-specific factors," *ACL*, 2018.

[75] H. Lu, K. N. Plataniotis, and A. Venetsanopoulos, "Multilinear subspace learning: Dimensionality reduction of multidimensional data," 2013.

[76] H. Lu, K. N. Plataniotis, and A. N. Venetsanopoulos, "Mpca: multilinear principal component analysis of tensor objects," *IEEE Transactions on Neural Networks*, vol. 19, no. 1, pp. 18–39, 2008.

[77] W. Luiz, F. Viegas, R. Alencar, F. Mourão, T. Salles, D. Carvalho, M. A. Gonçalves, and L. Rocha, "A feature-oriented sentiment rating for mobile app reviews," in *Proceedings of the 2018 World Wide Web Conference on World Wide Web*. International World Wide Web Conferences Steering Committee, 2018, pp. 1909–1918.

[78] L. v. d. Maaten and G. Hinton, "Visualizing data using t-sne," *Journal of machine learning research*, vol. 9, no. Nov, pp. 2579–2605, 2008.

[79] G. Mai, K. Cao, P. C. Yuen, and A. K. Jain, "On the reconstruction of face images from deep face templates," *IEEE transactions on pattern analysis and machine intelligence*, vol. 41, no. 5, pp. 1188–1202, 2018.

[80] X. Mao, Q. Li, H. Xie, R. Y. Lau, Z. Wang, and S. Paul Smolley, "Least squares generative adversarial networks," in *Proceedings of the IEEE International Conference on Computer Vision*, 2017, pp. 2794–2802.

[81] N. McLaughlin, J. M. Del Rincon, and P. Miller, "Data-augmentation for reducing dataset bias in person re-identification," in *2015 12th IEEE International conference on advanced video and signal based surveillance (AVSS)*. IEEE, 2015, pp. 1–6.

[82] A. K. Menon and R. C. Williamson, "The cost of fairness in binary classification," in *Conference on Fairness, Accountability and Transparency*, 2018, pp. 107–118.

[83] L.-P. Morency, R. Mihalcea, and P. Doshi, "Towards multimodal sentiment analysis: Harvesting opinions from the web," in *Proceedings of the 13th international conference on multimodal interfaces*. ACM, 2011, pp. 169–176.

[84] E. Morvant, A. Habrard, and S. Ayache, "Majority vote of diverse classifiers for late fusion," in *Joint IAPR International Workshops on Statistical Techniques in Pattern Recognition (SPR) and Structural and Syntactic Pattern Recognition (SSPR)*. Springer, 2014, pp. 153–162.

[85] J. Ngiam, Z. Chen, D. Chia, P. W. Koh, Q. V. Le, and A. Y. Ng, "Tiled convolutional neural networks," in *Advances in Neural Information Processing Systems*, 2010, pp. 1279–1287.

[86] S. Park, H. S. Shim, M. Chatterjee, K. Sagae, and L.-P. Morency, "Computational analysis of persuasiveness in social multimedia: A novel dataset and multimodal prediction approach," in *Proceedings of the 16th International Conference on Multimodal Interaction*. ACM, 2014.

[87] N. Passalis and A. Tefas, "Training lightweight deep convolutional neural networks using bag-of-features pooling," *IEEE transactions on neural networks and learning systems*, 2018.

[88] M. Paulin, J. Revaud, Z. Harchaoui, F. Perronnin, and C. Schmid, "Transformation pursuit for image classification," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2014, pp. 3646–3653.

[89] J. Pennington, R. Socher, and C. Manning, "Glove: Global vectors for word representation," in *Proceedings of the 2014 conference on empirical methods*

in natural language processing (EMNLP), 2014, pp. 1532–1543.

[90] V. Pérez-Rosas, R. Mihalcea, and L.-P. Morency, "Utterance-level multimodal sentiment analysis," in *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 1)*, 2013, pp. 973–982.

[91] S. Poria, E. Cambria, and A. Gelbukh, "Deep convolutional neural network textual features and multiple kernel learning for utterance-level multimodal sentiment analysis," in *Proceedings of the 2015 conference on empirical methods in natural language processing*, 2015, pp. 2539–2544.

[92] S. Poria, E. Cambria, D. Hazarika, N. Majumder, A. Zadeh, and L.-P. Morency, "Context-dependent sentiment analysis in user-generated videos," in *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 2017.

[93] S. Poria, E. Cambria, N. Howard, G.-B. Huang, and A. Hussain, "Fusing audio, visual and textual clues for sentiment analysis from multimodal content," *Neurocomputing*, vol. 174, pp. 50–59, 2016.

[94] S. Rabanser, S. Günnemann, and Z. C. Lipton, "Failing loudly: An empirical study of methods for detecting dataset shift," *arXiv preprint arXiv:1810.11953*, 2018.

[95] A. Radford, L. Metz, and S. Chintala, "Unsupervised representation learning with deep convolutional generative adversarial networks," *ICLR*, 2016.

[96] G. A. Ramirez, T. Baltrušaitis, and L.-P. Morency, "Modeling latent discriminative dynamic of multi-dimensional affective signals," in *International Conference on Affective Computing and Intelligent Interaction*. Springer, 2011, pp. 396–406.

[97] J. Ren, Y. Hu, Y.-W. Tai, C. Wang, L. Xu, W. Sun, and Q. Yan, "Look, listen and learna multimodal lstm for speaker identification," in *Thirtieth AAAI Conference on Artificial Intelligence*, 2016.

[98] S. Rendle, "Factorization machines," in *Data Mining (ICDM), 2010 IEEE 10th International Conference on*. IEEE, 2010s, pp. 995–1000.

[99] S. Rifai, P. Vincent, X. Muller, X. Glorot, and Y. Bengio, "Contractive auto-encoders: explicit invariance during feature extraction," in *Proceedings of the 28th International Conference on International Conference on Machine Learning*. Omnipress, 2011, pp. 833–840.

[100] T. Salimans, I. Goodfellow, W. Zaremba, V. Cheung, A. Radford, and X. Chen, "Improved techniques for training gans," in *Advances in Neural Information Processing Systems*, 2016, pp. 2234–2242.

[101] I. Sato, H. Nishimura, and K. Yokoi, "Apac: Augmented pattern classification with neural networks," *arXiv preprint arXiv:1505.03229*, 2015.

[102] B. Savas and L. Eldén, "Handwritten digit classification using higher order singular value decomposition," *Pattern Recognition*, vol. 40, no. 3, pp. 993–1003, 2007.

[103] S. T. Shivappa, M. M. Trivedi, and B. D. Rao, "Audiovisual information fusion in human–computer interfaces and intelligent environments: A survey," *Proceedings of the IEEE*, vol. 98, no. 10, pp. 1692–1715, 2010.

[104] K. Shmelkov, C. Schmid, and K. Alahari, "How good is my gan?" in *Proceedings of the European Conference on Computer Vision (ECCV)*, 2018, pp. 213–229.

[105] A. Shrivastava, T. Pfister, O. Tuzel, J. Susskind, W. Wang, and R. Webb,

"Learning from simulated and unsupervised images through adversarial training," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 2242–2251.

[106] K. Sohn and H. Lee, "Learning invariant representations with local transformations," in *Proceedings of the 29th International Coference on International Conference on Machine Learning*, 2012, pp. 1339–1346.

[107] K. Sohn, G. Zhou, C. Lee, and H. Lee, "Learning and selecting features jointly with point-wise gated boltzmann machines," in *International Conference on Machine Learning*, 2013, pp. 217–225.

[108] L. Sorber, M. Van Barel, and L. De Lathauwer, "Structured data fusion," *IEEE Journal of Selected Topics in Signal Processing*, vol. 9, no. 4, pp. 586–600, 2015.

[109] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Dropout: a simple way to prevent neural networks from overfitting," *The Journal of Machine Learning Research*, vol. 15, no. 1, pp. 1929–1958, 2014.

[110] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich, "Going deeper with convolutions," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2015, pp. 1–9.

[111] T. Tommasi, N. Patricia, B. Caputo, and T. Tuytelaars, "A deeper look at dataset bias," in *Domain Adaptation in Computer Vision Applications*. Springer, 2017, pp. 37–55.

[112] A. Torralba and A. A. Efros, "Unbiased look at dataset bias," in *Computer Vision and Pattern Recognition, IEEE Conference on*, 2011.

[113] L. R. Tucker, "The extension of factor analysis to three-dimensional matrices," *Contributions to Mathematical Psychology*, vol. 110119, 1964.

[114] I. Van Mechelen and A. K. Smilde, "A generic linked-mode decomposition model for data fusion," *Chemometrics and Intelligent Laboratory Systems*, vol. 104, no. 1, pp. 83–94, 2010.

[115] M. Varma and B. R. Babu, "More generality in efficient multiple kernel learning," in *Proceedings of the 26th Annual International Conference on Machine Learning*. ACM, 2009, pp. 1065–1072.

[116] M. Varma and A. Zisserman, "A statistical approach to material classification using image patch exemplars," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 31, no. 11, pp. 2032–2047, 2009.

[117] M. A. O. Vasilescu and D. Terzopoulos, "Multilinear image analysis for facial recognition," in *Pattern Recognition, 2002. Proceedings. 16th International Conference on*, vol. 2. IEEE, 2002, pp. 511–514.

[118] S. Verma, W. Liu, C. Wang, and L. Zhu, "Extracting highly effective features for supervised learning via simultaneous tensor factorization." in *AAAI*, 2017, pp. 4995–4996.

[119] ——, "Hybrid networks: Improving deep learning networks via integrating two views of images," in *Neural Information Processing - 25th International Conference, ICONIP , Proceedings, Part I*, 2018, pp. 46–58. [Online]. Available: https://doi.org/10.1007/978-3-030-04167-0_5

[120] K. Wang, C. Gou, Y. Duan, Y. Lin, X. Zheng, and F.-Y. Wang, "Generative adversarial networks: introduction and outlook," *IEEE/CAA Journal of Automatica Sinica*, vol. 4, no. 4, pp. 588–598, 2017.

[121] S. Wang, M. Zhou, G. Fei, Y. Chang, and B. Liu, "Contextual and position-aware factorization machines for sentiment classification," *arXiv preprint arXiv:1801.06172*, 2018.

[122] Y. Wang, A. Sun, J. Han, Y. Liu, and X. Zhu, "Sentiment analysis by capsules," in *Proceedings of the 2018 World Wide Web Conference on World Wide Web*. International World Wide Web Conferences Steering Committee, 2018, pp. 1165–1174.

[123] Y. Xian, T. Lorenz, B. Schiele, and Z. Akata, "Feature generating networks for zero-shot learning," *CVPR*, 2018.

[124] L. Yang, D. Jiang, and H. Sahli, "Integrating deep and shallow models for multi-modal depression analysis—hybrid architectures," *IEEE Transactions on Affective Computing*, 2018.

[125] Z. Yang, J. Hu, R. Salakhutdinov, and W. W. Cohen, "Semi-supervised qa with generative domain-adaptive nets," *arXiv preprint arXiv:1702.02206*, 2017.

[126] T. Young, D. Hazarika, S. Poria, and E. Cambria, "Recent trends in deep learning based natural language processing," *ieee Computational intelligenCe magazine*, vol. 13, no. 3, pp. 55–75, 2018.

[127] W. Yu, H. Zhang, X. He, X. Chen, L. Xiong, and Z. Qin, "Aesthetic-based clothing recommendation," in *Proceedings of the 2018 World Wide Web Conference*. International World Wide Web Conferences Steering Committee, 2018, pp. 649–658.

[128] J. Yuan and M. Liberman, "Speaker identification on the scotus corpus," *Journal of the Acoustical Society of America*, vol. 123, no. 5, p. 3878, 2008.

[129] A. Zadeh, M. Chen, S. Poria, E. Cambria, and L.-P. Morency, "Tensor fusion network for multimodal sentiment analysis," in *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, 2017, pp. 1103–1114.

[130] A. Zadeh, P. P. Liang, N. Mazumder, S. Poria, E. Cambria, and L.-P. Morency, "Memory fusion network for multi-view sequential learning," in *Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence, New Orleans, Louisiana, USA, February 2-7, 2018*, 2018.

[131] A. Zadeh, P. P. Liang, S. Poria, P. Vij, E. Cambria, and L.-P. Morency, "Multi-attention recurrent network for human communication comprehension," in *Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence, New Orleans, Louisiana, USA, February 2-7, 2018*, 2018.

[132] A. Zadeh, R. Zellers, E. Pincus, and L.-P. Morency, "Multimodal sentiment intensity analysis in videos: Facial gestures and verbal messages," *IEEE Intelligent Systems*, vol. 31, no. 6, pp. 82–88, 2016.

[133] X. Zhang, J. Zou, X. Ming, K. He, and J. Sun, "Efficient and accurate approximations of nonlinear convolutional networks," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2015, pp. 1984–1992.

[134] L. Zheng, Y. Yang, and Q. Tian, "Sift meets CNN: a decade survey of instance retrieval," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 40, no. 5, pp. 1224–1244, 2018.

[135] J.-Y. Zhu, T. Park, P. Isola, and A. A. Efros, "Unpaired image-to-image translation using cycle-consistent adversarial networks," in *ICCV*, 2017.

[136] J.-Y. Zhu, R. Zhang, D. Pathak, T. Darrell, A. A. Efros, O. Wang, and E. Shechtman, "Toward multimodal image-to-image translation," in *Advances in Neural Information Processing Systems*, 2017, pp. 465–476.

[137] Q. Zhu, M.-C. Yeh, K.-T. Cheng, and S. Avidan, "Fast human detection using a cascade of histograms of oriented gradients," in *Computer Vision and Pattern Recognition, 2006 IEEE Conference on*, vol. 2. IEEE, 2006, pp. 1491–1498.