ABSTRACT

Title of dissertation: DEEP LEARNING WITH CONSTRAINTS AND PRIORS FOR IMPROVED SUBJECT CLUSTERING, MEDICAL IMAGING, AND ROBUST INFERENCE

Wei-An Lin
Doctor of Philosophy, 2020

Dissertation directed by: Professor Rama Chellappa
Department of Electrical and Computer Engineering

Deep neural networks (DNNs) have achieved significant success in several fields including computer vision, natural language processing, and robot control. The common philosophy behind these success is the use of large amount of annotated data and end-to-end networks with task-specific constraints and priors implicitly incorporated into the trained model without the need for careful feature engineering. However, DNNs are shown to be vulnerable to distribution shifts and adversarial perturbations, which indicates that such implicit priors and constraints are not sufficient for real world applications. In this dissertation, we target three applications and design task-specific constraints and priors for improved performance of deep neural networks.

We first study the problem of subject clustering, the task of grouping face images of the same person together. We propose to utilize the prior structure in the feature space of DNNs trained for face identification to design a novel clustering algorithm. Specifically, the clustering algorithm exploits the local neighborhood

structure of deep representations by exemplar-based learning based on k-nearest neighbors (k-NN). Extensive experiments show promising results for grouping face images according to subject identity. As an example, we apply the proposed clustering algorithm to automatically curate a large-scale face dataset with noisy labels and show that the performance of face recognition DNNs can be significantly improved by training on the curated dataset. Furthermore, we empirically find that the k-NN rule does not capture proper local structures for deep representations when each subject has very few face images. We then propose to improve upon the exemplar-based approach by a density-aware similarity measure and theoretically show its asymptotic convergence to a density estimator. We conduct experiments on challenging face datasets that show promising results.

Second, we study the problem of metal artifact reduction in computed tomography (CT). Unlike typical image restoration tasks such as super-resolution and denoising, metal artifacts in CT images are structured and non-local. Conventional DNNs do not generalize well when metal implants with unseen shapes are presented. We find that the imaging process of CT induces a data consistency prior that can be exploited for image enhancement. Based on this observation, we propose a dual-domain learning approach to CT metal artifact reduction. We design and implement a novel Radon inversion layer that allows gradients in the image domain to be back-propagated to the projection domain. Experiments conducted on both simulated datasets and clinical datasets show promising results. Compared to conventional DNN-based models, the proposed dual-domain approach leads to impressive metal artifact reduction and has improved generalization capability.

Finally, we study the problem of robust classification. In the past few years, the vulnerability of DNNs to small imperceptible perturbations has been widely studied, which raises concerns about the security and robustness of DNNs against possible threat models. To defend against threat models, Samangoui *et al.* proposed DefenseGAN, a preprocessing approach which removes adversarial perturbations by projecting the input images onto the learned data prior. However, the projection operation in DefenseGAN is time-consuming and may not yield proper reconstruction when images have complicated textures. We propose an inversion network to constrain the initial estimates of the latent code for input images. With the proposed constraint, the number of optimization steps in DefenseGAN can be reduced while achieving improved accuracy and robustness. Furthermore, we conduct empirical studies on attack methods that have claimed to break DefenseGAN, which shows that on-manifold robustness might be the key factor for ensuring adversarial robustness.

DEEP LEARNING WITH CONSTRAINTS AND PRIORS FOR IMPROVED SUBJECT CLUSTERING, MEDICAL IMAGING, AND ROBUST INFERENCE

by

Wei-An Lin

Dissertation submitted to the Faculty of the Graduate School of the
University of Maryland, College Park in partial fulfillment
of the requirements for the degree of
Doctor of Philosophy
2020

Advisory Committee:
Professor Rama Chellappa, Chair/Advisor
Professor Behtash Babadi
Professor Gang Qu
Professor Min Wu
Professor Ramani Duraiswami, Dean's Representative

Acknowledgments

I can still clearly remember five years ago, when I decided to switch my research direction from wireless communication to computer vision, and when I first stepped into Professor Rama Chellappa's office, introducing myself and starting a new journey in this large research group. Looking back from now, I've experienced and learned so many things. I'm thankful to all the people I met in this journey. Without you, I couldn't have survived this challenge.

First of all, I would like to thank my advisor Professor Rama Chellappa. He is the most supportive and humorous professor I've ever met. Whenever I was about to give up on something, I remembered his witty remark: "I see people get married for several times, but I only see people study for one PhD." Professor Chellappa always uses his unique sense of humor to inspire and motivate his students. I really appreciate the opportunity to work in this large group and advised by Professor Chellappa.

Among all my teammates in this large family, I would like to especially thank Dr. Jun-Cheng Chen for his constructive advice and helpful guidance. Dr. Chen and I co-worked several papers in this dissertation. I always learned a lot from the discussions and collaborations with him.

I would also like to thank Professor Babadi, Professor Wu, Professor Qu, and Professor Duraiswami for serving as my dissertation committee members. They provided valuable comments to this dissertation and guidance for possible future research directions.

I'm also grateful for having the opportunity to work as an intern with Dr. Kevin Zhou. The experience I gained during the internship drastically enriched my PhD research. I would also like to thank the collaborators Haofu Liao and Cheng Peng. I learned a lot from their creative ideas, diligence, and enthusiasms for research.

I cannot be more thankful to my family. Without them, these achievements would not be possible. In particular, I owe my deepest thanks to my mother for her encouragement and unconditional love.

Finally, part of my research is based upon work supported by the Office of the Director of National Intelligence (ODNI), Intelligence Advanced Research Projects Activity (IARPA), via IARPA R&D Contract No. 2014-14071600012. The views and conclusions contained herein are those of the authors and should not be interpreted as necessarily representing the official policies or endorsements, either expressed or implied, of the ODNI, IARPA, or the U.S. Government. The U.S. Government is authorized to reproduce and distribute reprints for Governmental purposes notwithstanding any copyright annotation thereon.

Table of Contents

List of Figures

List of Tables

Chapter 1: Introduction

1.1 Motivation

Deep learning has achieved significant success in several fields including computer vision, natural language processing, and robot control. The common philosophy behind these success is the use of large amount of annotated data and end-to-end networks with proper architectural design, which enforces domain-specific constraints and priors into the learning model while requiring a minimal amount of feature engineering. The most successful example is the development of convolutional neural networks (CNNs) for images and videos. CNNs consist of a set of convolutional, nonlinear, pooling, and fully connected layers, which resembles human visual cortex system. Mathematically, the translation-equivariant property for convolution operation is suitable for processing natural images and videos. CNNs have shown to be extremely successful in low-level visual tasks such as super-resolution, deblurring, and high-level visual tasks such as classification, segmentation, and understanding. Another example is the well-developed recurrent neural networks (RNNs) in sequential data modeling. Unlike CNNs which treat each data independently, RNNs use hidden state to model the dependency between data samples. RNNs have been widely adopted for audio, natural language, and time sequence processing. Recently,

the work of deep image prior (DIP) [102] shows that the network architecture itself is a suitable prior for image processing tasks such as super-resolution, denoising, and inpainting.

Despite these successes, DNNs are shown to be vulnerable under distribution shift, i.e. DNNs do not generalize well when the test domain has a different distribution from the training domain, and adversarial examples, i.e. small perturbations in the input can lead to undesirable changes in the output. A pioneering study by Ilyas *et al.* [40] found out that DNNs trained with empirical risk minimization tend to exploit non-robust features in images to achieve high accuracy, which indicates that the priors and constraints enforced by the domain-specific architectural designs are not sufficient in order to build reliable systems for real world applications. In this dissertation, we aim to address these problems by designing deep learning constraints and priors for improved generalization and adversarial robustness.

1.2 Overview

The idea of imposing constraints and priors to regularize the learning model can be traced back to the classical Bayesian inference approach. In the Bayesian approach, the prior knowledge about the unknown parameters of interest is specified by a probability distribution. After acquiring new data, the belief about the parameters is updated via Bayes' Theorem. To ensure tractability and applicability, prior distributions are often selected to have simple analytical forms. With the development of deep learning, constraints and priors can be purely empirical and

complicated. For example, generative adversarial networks (GANs), an empirical model capturing the distribution of natural images, can be used as an image prior to perform tasks including super-resolution, colorization and inpainting without any supervision [5, 32]. Works in image editing [17, 46, 124] show that imposing similarity constraints in deep feature space leads to superior performance than conventional L1/L2 constraints in image space.

In this dissertation, we target three applications for which we design deep learning constraints and priors for improved generalization and adversarial robustness. In Chapters 2 and 3, we study the problem of subject clustering: grouping face images of the same person together. Since DNNs trained for face identification induce a special structure for the extracted features, we propose to use this prior knowledge in the design of novel clustering algorithms. Specifically, we present Proximity-Aware Hierarchical Clustering (PAHC) for unconstrained faces. The clustering algorithm exploits the local structure of deep representations and hence has improved capability for grouping face images according to subject identity. As a demonstrative example, we apply PAHC to automatically curate a large-scale face dataset, i.e. MS-Celeb-1M, and show that the performance of face recognition DNNs can be significantly improved by training on the curated dataset.

In Chapter 4, we study the problem of metal artifact reduction in computed tomography (CT). Unlike typical image restoration tasks such as super-resolution and denoising, metal artifacts in CT images are structured and non-local. Conventional DNNs do not generalize well when metal implants with unseen shapes are presented. To address this issue, we show the imaging process of CT induces a

data consistency prior that can be exploited for image enhancement. Based on this finding, we propose the Dual Domain Network (DuDoNet) for CT metal artifact reduction. In DuDoNet, we introduce a novel Radon inversion layer which uses data from the projection domain to regularize image domain enhancement. The proposed dual domain approach leads to impressive metal artifact reduction.

In Chapter 5, we study the problem of robust classification. In the past few years, the vulnerability of DNNs to small imperceptible perturbations has been widely studied. To defend against the threat models, Samangoui *et al.* proposed DefenseGAN, a preprocessing approach which removes adversarial perturbations. We improve upon DefenseGAN by finding an inversion network to provide initial estimates to the latent code of input images. With the initial estimates, the number of optimization steps in DefenseGAN can be reduced while achieving improved accuracy and robustness. Furthermore, we conduct empirical studies on attack methods that have claimed to break DefenseGAN partially (55% accuracy on MNIST) or completely (3% accuracy on MNIST).

1.3 Contributions

- In Chapter 2, we propose an exemplar-based learning approach for face subject clustering.
  - We present a similarity measure based on exemplar-based large-margin classification. With the proposed similarity measure, clusters are formed by applying hierarchical clustering.

  - We show the proposed method yields promising empirical results.
  - We apply the proposed clustering algorithm to automatically curate a large-scale dataset with noisy labels. DNNs trained on the curated dataset achieve significant performance improvement.

- In Chapter 3, we propose a density-based similarity measure for face subject clustering.
  - We propose to measure the similarity between deep features by a density-aware measure. Clusters are then formed by the hierarchical clustering algorithm.
  - We mathematically show that the proposed similarity measure asymptotically converges to a density estimator.
  - We conduct extensive experiments and show that the proposed density-aware clustering method outperforms baseline approaches.

- In Chapter 4, we propose a dual-domain learning approach for CT metal artifact reduction.
  - We propose an end-to-end trainable dual-domain refinement network for metal artifact reduction. The network is able to recover details corrupted by metal artifacts.
  - We propose a Radon inversion layer to enable efficient end-to-end dual-domain learning.
  - We propose a Radon consistency loss to penalize secondary artifacts in the image domain. Gradients of the loss in the image domain are back-

propagated to the sinogram domain for improved consistency.

  - Experimental evaluations conducted on CT images with simulated and real metal artifacts show the proposed dual-domain learning approach achieves superior performance against baseline methods.

- In Chapter 5, we propose an inversion algorithm to improve the efficiency and robustness of DefenseGAN.

  - We propose to train an inversion network to constrain the initial estimates of the latent code for input images.
  - We propose to improve the attack detection accuracy of DefenseGAN by using the semantic distance instead of the image space L2 distance.
  - We show that the proposed inversion method improves the efficiency of DefenseGAN.
  - We evaluate attacks that claim to break DefenseGAN, and show they are either not successful or some flaws exist when crafting adversarial examples.

Chapter 2: Proximity-Aware Hierarchical Clustering of Faces

2.1 Overview

In this chapter, we propose a face clustering algorithm called "Proximity-Aware Hierarchical Clustering" (PAHC) that exploits the local structure of deep representations. In the proposed method, a similarity measure between deep features is computed by evaluating linear SVM margins, which are learned using nearest neighbors of sample data. Clusters are then formed by applying agglomerative hierarchical clustering (AHC). We evaluate the clustering performance using four unconstrained face datasets, including Celebrity in Frontal-Profile (CFP), Labeled Faces in the Wild (LFW), IARPA JANUS Benchmark A (IJB-A), and IARPA JANUS Benchmark B (IJB-B) datasets. Experimental results demonstrate that the proposed approach achieves improved performance over state-of-the-art methods. Moreover, we show that the proposed clustering algorithm has the potential to actively learn robust deep face representations by first harvesting sufficient number of unseen face images through curation of a large-scale dataset, e.g. the MS-Celeb-1M dataset. By training DNNs on the curated MS-Celeb-1M dataset which contains over three million face images, improved representations for face images are learned.

2.2 Introduction

In this chapter, we address the problem of face clustering, especially for the scenario of grouping a set of face images without knowing the exact number of clusters. Face clustering algorithms provide meaningful partitions for given face image sets by combining faces with similar appearances while separating dissimilar ones. Ideally, face images in a partition should belong to the same identity, while images from different partitions should not. Identity-sensitive face clustering is an active research area in computer vision with several applications, including but not limited to organizing personal pictures, summarizing images from social media, and surveillance applications. Clustering is also important when training a data-hungry deep convolutional neural network (e.g. DenseNet [39] or ResNet [36]) for face verification, classification, or detection tasks. Recently, Microsoft Research released the MS-Celeb-1M dataset [34], which contains 1M celebrity names and over 8 million face images. Due to its diversity, this very-large dataset has the potential to improve the performance of face recognition systems. However, since the MS-Celeb-1M dataset has been built from the outputs of search engines, labeling errors could adversely affect the training of deep networks. An effective approach to tackle this problem is to apply a reliable clustering algorithm on the MS-Celeb-1M training dataset to harvest sufficient number of face images.

Despite extensive studies on general clustering algorithms over the past few decades, face image clustering remains to be a difficult task. The difficulties are mainly two-fold. Since face images of a person may have large variations in illu-

mination, facial expressions, occlusion, age, and pose, it is challenging to measure the similarity between two face images. The other issue is that without knowing the actual number of clusters, many well-established clustering algorithms, such as k-means, may not be effective.

To address these problems, we first apply a DNN to extract deep features from given face images, which are robust to face variations. We then define a novel similarity measure that is aware of local information. Based on the similarity measure, AHC is applied to yield face clusters. Unlike [118], our approach does not require the exact number of clusters as a prior and repeated training of a DNN. Unlike in [73] where a similarity measure between two points is computed only through the presence or absence of nearest neighbors, our approach measures the similarity between neighborhoods directly in the feature space: neighborhood geometries are first transferred to an evaluation hyperplane, pairwise similarity is then obtained by evaluating the points on the hyperplanes.

This work builds upon [60]. In addition to the evaluations presented in [60], we further study one application of the proposed method in active learning. Specifically, we first curate the MS-Celeb-1M dataset using the proposed PAHC algorithm with a light-weight DCNN. We then train two different DCNNs on the curated MS-Celeb-1M for improved face representations. We also carry out extensive experiments on the LFW and IJB-B datasets and gain deeper insights on how the proposed algorithm performs when the given data has very distinct distributions.

The rest of the chapter is organized as follows. We summarize related works in Section 2.3. The proposed algorithm is detailed in Section 2.4. In Section 2.5, we

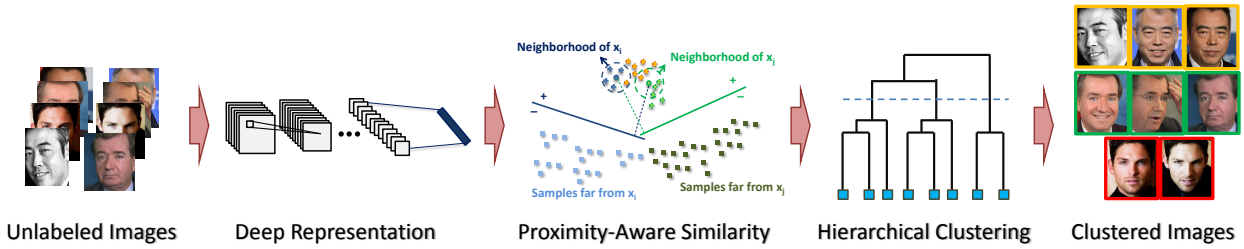


Figure 2.1: Overall pipeline for the proposed PAHC algorithm. Unlabeled face images are preprocessed and passed through a DNN to obtain deep features. The Proximity-Aware similarity between each pair of features is then computed. Based on the Proximity-Aware similarity, hierarchical clustering is applied to yield the final image clusters.

carry out qualitative and quantitative evaluations and demonstrate the effectiveness of the proposed approach. Finally, conclusions are given in Section 2.6.

2.3 Backgrounds and Related Works

2.3.1 Deep Representation

Recent advances related to DNNs have brought about impressive improvements for image classification and verification tasks [53, 95], which can be attributed to their ability to extract discriminative information from each image and represent it compactly. DCNNs trained on labeled face images have been used in [91, 98] for face recognition tasks. Inspired by these advances, we apply a DCNN to extract deep features from the given faces that retain sufficient amount of information to distinguish among different identities.

2.3.2 Conventional Clustering Algorithms

Clustering algorithms can be generally categorized into partitioning and agglomerative approaches. Both approaches build upon a similarity graph $G(V, E)$ defined for the given data points. The graph can be either fully connected, in ϵ-neighborhood or in k-nearest neighbors. For partitioning approaches, given the number of clusters, k-means [64] iteratively updates the group centers and corresponding members until convergence. Spectral clustering finds the underlying structure based on graph Laplacian [72, 93, 121]. For agglomerative approaches [31, 55], groups of data points are merged whenever the linkage between them is above some threshold. Finding the proper similarity measure is one of the major tasks in designing clustering algorithms. Traditional approaches use non-increasing functions of pairwise distance as the similarity measure, *e.g.* $\exp(-d(\mathbf{x}_i, \mathbf{x}_j)^2/\sigma^2)$.

2.3.3 Sparse Subspace Clustering

Recently, sparse subspace clustering (SSC) [20, 21] and low-rank subspace clustering (LRSC) [62, 104], which exploit the subspace structures in a dataset, have gained significant attention. Both methods assume data points have low-dimensional structures. By minimizing the reconstruction error under the sparsity/low-rank criterion, the similarity matrix can be obtained from the corresponding sparse/low-rank representation. However, SSC and LRSC are computationally expensive and do not scale well. In [78], dimensionality reduction and subspace clustering are simultaneously learned to achieve improved performance and efficiency. In [120],

high clustering performance is achieved on the Extended Yale B dataset, which contains face images in controlled variations. However, in unconstrained settings, face images no longer have low-dimensional structure, making SSC ineffective.

2.3.4 Image Clustering based on Deep Neural Networks

Yang *et al.* [118] proposed learning deep representations and image clusters jointly in a recurrent framework. Each image is treated as a separate cluster at the beginning, and a deep network is trained using this initial grouping. The deep representation and cluster members are then iteratively refined until the number of clusters reached the predefined value. Zhang *et al.* [129] proposed to cluster face images in videos by alternating between deep representation adaption and clustering. Temporal and spatial information between and within video frames is exploited to achieve high purity face image clusters. Otto *et al.* [73] proposed the Approximate Rank-Order clustering algorithm that modifies the algorithm in [132] by (i) using deep representations of images (ii) considering only the absence and presence of the shared nearest neighbors and (iii) transitively merging only once. Superior clustering results and computational time are achieved from the modifications. In [94], the authors formulated the clustering problem as solving a conditional random field model built upon deep representations. The proposed ConPaC algorithm outperforms Approximate Rank-Order on many datasets.

Different from these studies, we propose a clustering algorithm that does not require (i) training a deep network iteratively [118] and (ii) partial identity informa-

tion [129]. Our approach focuses on exploiting the neighborhood structure between samples and implicitly performs domain adaptation to achieve improved clustering performance.

2.3.5 Domain Adaptation

Domain adaptation aims at transferring features learned in the source domain to some unseen target domain. In the context of subject clustering, domain adaptation algorithms can be applied to learn reliable feature representations for unseen subjects. However, most of the existing methods [25, 65, 86, 101] target closed set domain adaptation, where both source and target domains contain the same classes. In subject clustering, unsupervised open set domain adaptation [10], where source and target domains only share a few (or no) categories, should be considered.

2.4 Proposed Method

In this section, we introduce our clustering algorithm, illustrated in Figure 2.1. The face images first pass through a pre-trained face DCNN model to extract deep features. Then, we compute the Proximity-Aware similarity scores using linear SVMs trained with corresponding neighborhoods of the samples. Finally, AHC is applied on the similarity scores to determine the cluster labels to each sample. The details of each components are described in the following subsections.

2.4.1 Notation

We denote the set of face images as $I = \{I_1, \ldots, I_N\}$. Our goal is to assign labels $L = \{l_1, \ldots, l_N\}$ for each image to indicate the cluster it belongs to. The images are first passed through a pre-trained DCNN model to extract the deep features, which are then normalized to unit length. Specifically, let $f_\theta : \mathcal{I} \to \mathcal{X}$ be the DCNN network parameterized by θ, and $g : \mathcal{X} \to \mathcal{X}$ be the normalization. The corresponding deep representations for the face images are given by $X = g \circ f_\theta(I) = \{\mathbf{x}_1, \ldots, \mathbf{x}_N\}$. For each representation $\mathbf{x}_i$, we define $\mathcal{V}_K(\mathbf{x}_i)$ as the set of K-nearest neighbors of $\mathbf{x}_i$, including $\mathbf{x}_i$ itself.

2.4.2 Proximity-Aware Similarity

Recent advances in DCNN have yielded great improvements for face verification task, which uses cosine distance as the similarity measure to decide whether two faces belong to the same subject. Given two features $\mathbf{x}_i, \mathbf{x}_j \in \mathcal{X}$ on the unit hypersphere $\{\mathbf{x} : \|\mathbf{x}\| = 1\}$, the similarity between them is computed by

$$s(\mathbf{x}_i, \mathbf{x}_j) = \mathbf{x}_i^T \mathbf{x}_j. \tag{2.1}$$

The pairwise distance matrix $\mathbf{D}$ in this case is simply

$$[\mathbf{D}]_{i,j} = 1 - s(\mathbf{x}_i, \mathbf{x}_j). \tag{2.2}$$

Since DCNNs trained on large datasets extract discriminative features for images, distance measure based on (2.2) can be used to distinguish faces with distinct identities if they have similar distribution as the training dataset. However, the difference in distribution encountered in many real-world applications degrades the performance significantly. Inspired by previous works [73, 132], we measure similarity based on the neighborhood structure of deep features.

To have a formulation that is able to take neighborhoods $\mathcal{V}_K(\mathbf{x}_i)$, $\mathcal{V}_K(\mathbf{x}_j)$ into account when measuring the similarity between $\mathbf{x}_i$ and $\mathbf{x}_j$, we rewrite the inner product as

$$s(\mathbf{x}_i, \mathbf{x}_j) = \frac{\mathbf{x}_i^T \mathbf{x}_j + \mathbf{x}_j^T \mathbf{x}_i}{2}. \tag{2.3}$$

In (2.3), the similarity between $\mathbf{x}_i$ and $\mathbf{x}_j$ is evaluated by averaging two asymmetric measures: How similar is $\mathbf{x}_j$ from the view of $\mathbf{x}_i$ and how similar is $\mathbf{x}_i$ from the view of $\mathbf{x}_j$. Specifically, $\mathbf{x}_i^T \mathbf{x}_j$ can be interpreted as evaluating $\mathbf{x}_j$ on hyperplane $H_i = \{\mathbf{x} : \mathbf{x}_i^T \mathbf{x} = 0\}$ and $\mathbf{x}_j^T \mathbf{x}_i$ can be interpreted as evaluating $\mathbf{x}_i$ on hyperplane $H_j = \{\mathbf{x} : \mathbf{x}_j^T \mathbf{x} = 0\}$. This observation allows us to generalize the asymmetric measure as follows.

Given a hyperplane $H_{\mathbf{w}_i, b_i} = \{\mathbf{x} : \mathbf{w}_i^T \mathbf{x} + b_i = 0\}$ which contains information about $\mathcal{V}_K(\mathbf{x}_i)$, the asymmetric similarity from $H_{\mathbf{w}_i, b_i}$ to some set S is defined as

$$H_{\mathbf{w}_i, b_i}(S) = \frac{1}{|S|} \sum_{\mathbf{x} \in S} [\mathbf{w}_i^T \mathbf{x} + b_i]. \tag{2.4}$$

Following (2.3), the generalized similarity measure, which we call "Proximity-Aware

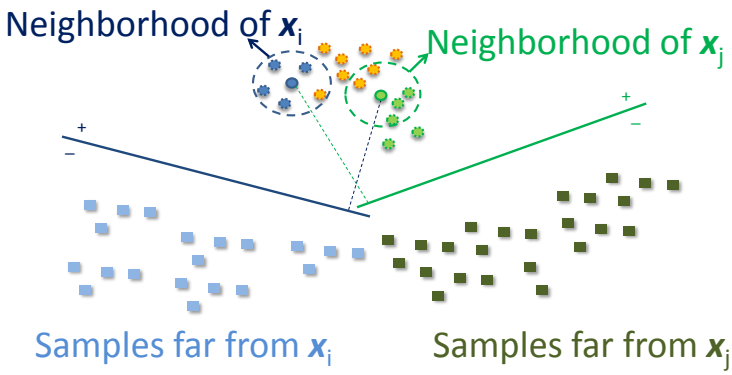


Figure 2.2: Proximity-Aware similarity. Circles in blue, yellow, green represent samples with different identities. Blue dashed circles delineate the neighborhood of $\mathbf{x}_i$ (or $\mathcal{V}_6(\mathbf{x}_i)$) while green dashed circles delineate the neighborhood of $\mathbf{x}_j$ (or $\mathcal{V}_6(\mathbf{x}_j)$). The blue hyperplane is obtained by solving (2.7), treating $\mathcal{V}_K(\mathbf{x}_i)$ as positive samples, and a subset of $X \backslash \mathcal{V}_K(\mathbf{x}_i)$, containing blue squares in this case, as negative samples. The green hyperplane is obtained in the same way. The Proximity-Aware similarity between $\mathbf{x}_i$ and $\mathbf{x}_j$ is evaluated using (2.5). The length of the blue dashed line and the green dashed line reflects how similar are the two neighborhoods.

similarity", is the average of two asymmetric measures from $H_{\mathbf{w}_i,b_i}$ to $\mathcal{V}_K(\mathbf{x}_j)$ and from $H_{\mathbf{w}_j,b_j}$ to $\mathcal{V}_K(\mathbf{x}_i)$:

$$s_{PA}(\mathbf{x}_i, \mathbf{x}_j) = \frac{H_{\mathbf{w}_i,b_i}(\mathcal{V}_K(\mathbf{x}_j)) + H_{\mathbf{w}_j,b_j}(\mathcal{V}_K(\mathbf{x}_i))}{2}. \tag{2.5}$$

Unlike cosine similarity, s_{PA} is not bounded. We introduce a nonlinear transformation to define the Proximity-Aware pairwise distance

$$[\mathbf{D}_{PA}]_{ij} = 1 - \frac{2}{\pi} \arctan\left[s_{PA}(\mathbf{x}_i, \mathbf{x}_j)\right]. \tag{2.6}$$

This choice of nonlinearity is for experimental simplicity. One can also consider $[\mathbf{D}_{PA}]_{i,j} = \exp(-s_{PA}(\mathbf{x}_i, \mathbf{x}_j))$. The above construction helps us to cast the problem of defining the similarity function between neighborhoods into finding hyperplanes

$H_{\mathbf{w}_i,b_i}$. Our ultimate goal is to find a similarity measure for each pair of feature vectors that reflects whether they belong to the same class. We conjecture that $H_{\mathbf{w}_i,b_i}$ and $H_{\mathbf{w}_j,b_j}$ should have the following property:

$H_{\mathbf{w}_i,b_i}(\cdot)$ has a large value when evaluating on sets that are near $\mathcal{V}_K(\mathbf{x}_i)$, and has a small value otherwise.

The constraint not only forces the similarity measure to be locally geometry-sensitive (proximity-aware) but also adaptive to the data domain. This justifies the use of linear classifiers to separate positive samples $\mathcal{V}_K(\mathbf{x}_i)$ from their corresponding negative samples. Figure 2.2 shows a demonstrative example. This approach is analogous to the one-shot similarity technique [110]. In this work, we use the linear SVM as our candidate algorithm for finding hyperplanes. Specifically, we solve

$$\begin{aligned}\min_{\mathbf{u}} \frac{1}{2}\mathbf{u}^T\mathbf{u} + C_p \sum_{k=1}^{N_p} \max[0, 1 - y_k \mathbf{u}^T \mathbf{z}_k]^2 \\ + C_n \sum_{k=1}^{N_n} \max[0, 1 - y_k \mathbf{u}^T \mathbf{z}_k]^2,\end{aligned} \tag{2.7}$$

where $\mathbf{u} = [\mathbf{w}^T \quad b]^T$ and $\mathbf{z}_k = [\mathbf{x}_k^T \quad 1]^T$. We treat $\mathcal{V}_K(\mathbf{x}_i)$ as positive samples with cardinality N_p, and a subset of $X \backslash \mathcal{V}_K(\mathbf{x}_i)$ as negative samples with cardinality N_n. $y_k = +1$ for positive samples and $y_k = -1$ for negative samples. The regularization constants C_p and C_n are given by $C_p = C\frac{N_p+N_n}{N_p}$ and $C_n = C\frac{N_p+N_n}{N_n}$.

In [110], Linear Discriminant Analysis (LDA) is used as the classifier to eval-

uate one-shot similarity score. However, we do not consider LDA as our candidate because the bimodal Gaussian prior assumption is not always satisfied for the positive and negative samples drawn from real-world datasets. In the proposed method, negative samples often consist of features from different identities with variations from nuisance factors, which do not obey a single Gaussian distribution.

2.4.3 Positive and Negative Sets Selection

The Proximity-Aware Similarity introduced in Section 2.4.2 is constructed by associating each data point $\mathbf{x}_i$ with its corresponding positive and negative samples. In this work, we propose to first construct the nearest neighbor list $\text{NNList}_{\mathbf{x}_i}$ for each data sample $\mathbf{x}_i$, where $\text{NNList}_{\mathbf{x}_i}[1] = \mathbf{x}_i$. The K-nearest neighbors of $\mathbf{x}_i$, $\mathcal{V}_K(\mathbf{x}_i)$ then corresponds to $\text{NNList}_{\mathbf{x}_i}[1:K]$. We select $\text{NNList}_{\mathbf{x}_i}[K':N]$ as the negative samples. Intuitively, the performance of the proximity-aware similarity depends on the true positive rate of positive samples and true negative rate of negative samples. In Section 2.5, we show how parameters (K, K') affect the clustering performance in detail.

2.4.4 Agglomerative Hierarchical Clustering

Agglomerative hierarchical clustering [31, 55] initializes all samples as separate clusters. Based on the pairwise distance matrix $\mathbf{D}$ measured from the features, clusters are iteratively merged whenever the cluster-to-cluster distance is below some threshold η. The hierarchical clustering algorithm, denoted as $\text{Hierarchical}(\mathbf{D}, \eta)$,

generates the cluster assignments L for all the faces in I. In our work, we use average linkage as a measure of cluster-to-cluster distance. Specifically, the average linkage between two clusters C_i and C_j can be computed by

$$d(C_i, C_j) = \frac{1}{|C_i|}\frac{1}{|C_j|}\sum_{u \in C_i}\sum_{v \in C_j} d(u, v). \tag{2.8}$$

The Proximity-Aware Hierarchical clustering is then characterized by the following algorithm:

$$L_{PA} \leftarrow \text{Hierarchical}(\mathbf{D}_{PA}, \eta). \tag{2.9}$$

2.5 Experimental Evaluations

In this section, we first show one application of the proposed algorithm in active learning. In Section 2.5.1, the PAHC algorithm is applied to curate the recently released MS-Celeb-1M [34] dataset. During the curation process, deep face representations are extracted by a light-weight DCNN [13] pretrained on CASIA-WebFace [119]. The CASIA-WebFace dataset contains 10,575 subjects and 494,414 images, which provides sufficient diversity and is in a reasonable scale for a DCNN with small *capacity* (*i.e.* number of parameters). In Section 2.5.2, we demonstrate how the curated large-scale dataset can aid in learning improved feature representations. Specifically, we train two DCNNs with different network capacities on the curated dataset, and show that both networks outperform the one trained on the smaller CASIA-WebFace dataset.

For the second part, we use the two DCNNs trained in Section 2.5.2 to carry

out extensive experiments on four datasets: LFW, CFP, IJB-A, and IJB-B. As shown in Figure 2.5, the four datasets have distinct variations in cluster sizes: CFP consists of uniform clusters, LFW consists of a large amount of singletons, IJB-A consists of clusters with diverse sizes, and IJB-B has the largest variations.

To compute Proximity-Aware similarity, we use the LIBLINEAR library [23] with L2-regularized L2-loss primal SVM. The parameter C is set at 10 throughout the experiments.

MS-Celeb-1M [34]:

Microsoft Research recently released this very large face image dataset, consisting of 1M identities. The training dataset of MS-Celeb-1M is prepared by selecting top 99,892 identities from the 1M celebrity list. There are 8,456,240 images in total, roughly 85 images per identity. This dataset is designed by leveraging a knowledge base called "freebase". Since face images are created using a search engine, labeling noise may be a problem when this dataset is used in supervised learning tasks. We demonstrate the effectiveness of the proposed clustering algorithm in curating large-scale noisy dataset in Section 2.5.1. We use aligned images provided with the dataset.

Celebrities in Frontal-Profile (CFP) [92]:

This dataset contains 500 subjects and 7,000 face images. Of the 7,000 faces, 5,000 are in frontal view, and the remaining 2,000 are in profile views where each subject contains 10 frontal and 4 profile images. Unlike the IJB-A dataset, the CFP

dataset aims at isolating the factor of pose variation in order to facilitate research in frontal-profile face verification. Extreme variations in poses can be seen in Figure 2.3. In this work, we apply our clustering algorithm on all 7,000 face images.

Labeled Faces in the Wild (LFW) [38]:

The dataset provides a set of unconstrained face images, which contains 13,233 images of 5,749 subjects. Since only faces that are detectable by the Viola-Jones detector are retained, the amount of variations in the dataset is limited. Note that 4,069 out of 5,749 subjects contain only one image.

IARPA Janus Benchmark A (IJB-A) [52]:

The IJB-A dataset contains 500 subjects with a total of 25,813 images taken from photos and video frames (5,399 still images and 20,414 video frames). Extreme variations in illumination, resolution, viewpoint, pose and occlusion make it a very challenging dataset. In this work, we cluster the templates corresponding to the query set for each split in IJB-A 1:1 verification protocol where a template is composed of a combination of still images and video frames. Figure 2.4 shows sample images from different templates.

IARPA Janus Benchmark B (IJB-B) [108]:

The IJB-B dataset is a superset of IJB-A. It consists of 1,845 subjects with 21,798 still images and 55,026 video frames. The diversity makes IJB-B even more challenging than IJB-A. The IJB-B dataset has a clustering protocol, which contains

seven subtasks. These subtasks differ in the number of distinct subjects, which involve 32, 64, 128, 256, 512, 1,024, and 1,845 subjects with a total of 1,026, 2,080, 5,224, 9,867, 18,251, 36,575, and 68,195 images, respectively.

Figure 2.3: Sample images in the CFP dataset. The first two rows are frontal face images and the last row consists of profile face images.

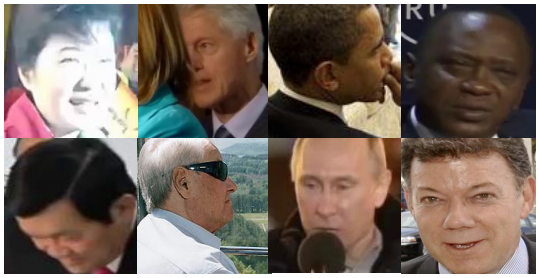
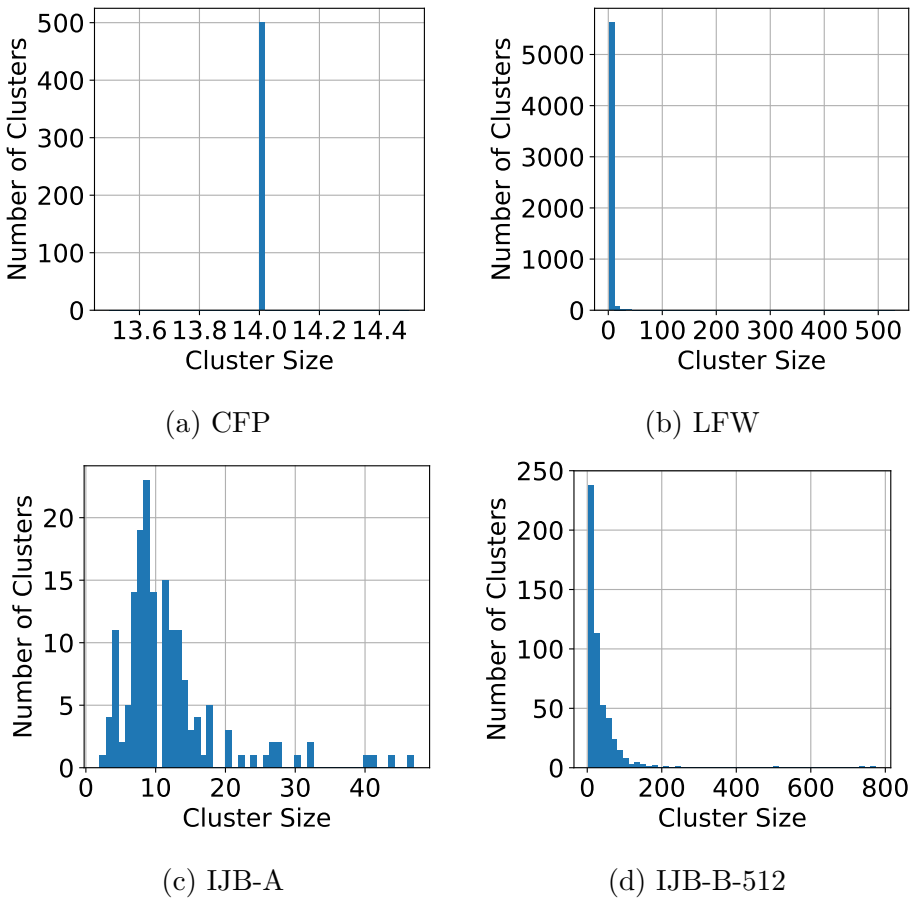


Figure 2.4: Sample images in IJB-A and IJB-B. The faces contain extreme illumination, viewpoint, pose, and occlusion changes.

2.5.1 Curation of the MS-Celeb-1M dataset

To extract deep features for all the face images, we first implement the DCNN presented in [13] and train it using the CASIA-WebFace dataset [119]. We denote this pretrained network as 'DNN (CASIA)'. DNN (CASIA) is trained using SGD for



(a) CFP (b) LFW

(c) IJB-A (d) IJB-B-512

Figure 2.5: Distribution of the size of clusters.

780K iterations with a standard batch size 128 and momentum 0.9. The learning rate is set to 1e-2 initially and is halved every 100K iterations. The weight decay rates of all the convolutional layers are set to 0, and the weight decay of the final fully connected layer is set to 5e-4.

We use the aligned images provided with the MS-Celeb-1M dataset. Feature representations are obtained by passing the whole dataset through DNN (CASIA). A total of 99,892 identities is divided into batches with size 50. For each batch, we apply Hierarchical($\mathbf{D}_{PA}, \eta = 2.3$), with $[\mathbf{D}_{PA}]_{i,j} = \exp(-s_{PA}(\mathbf{x}_i, \mathbf{x}_j))$, and s_{PA} is computed by $(K, K') = (5, 200)$. η, K, and K' are selected by cross-validating

on CASIA-WebFace. After PAHC, clusters whose majority identity have less than thirty images are discarded. After manually removing overlapped identities, the number of the curated dataset is about 3.7 millions face images of 57,440 identities. Figure 2.14 shows one example of the curated results. Since PAHC exploits local property, noisy labels are removed and sufficient amount of face images with extreme pose are retained.

2.5.2 Improved deep representation using the curated MS-Celeb-1M dataset

Deep neural networks have been shown to be successful in many machine learning tasks at the cost of large amount of carefully annotated data. In this section, we demonstrate that significant performance boost can be achieved by training on the large-scale dataset curated using PAHC and the light-weight DNN (CASIA). The result highlights an application of PAHC in collecting new training data for DCNNs without human labor. In the following, we first train two different DCNNs on the curated MS-Celeb-1M dataset, one with low and the other with high capacity. We then show the two trained DCNNs yield superior performance than DNN (CASIA) on the CFP and LFW datasets.

2.5.2.1 ResNet-101

We use ResNet-101 [36] network architecture for the task of face identification an verification. The model was trained on the 3.7 million images, containing $57,440$

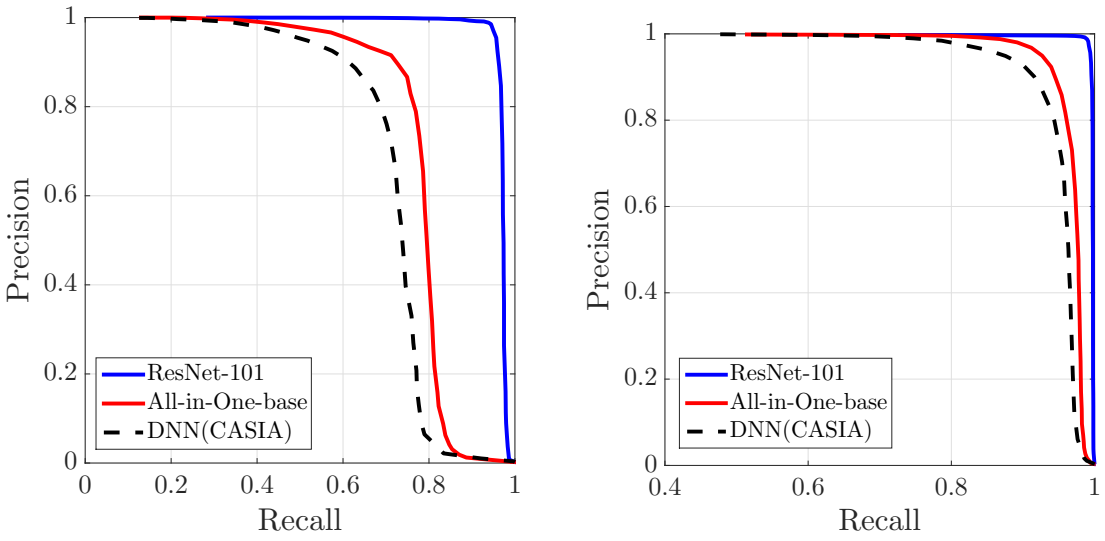
subjects from the curated dataset described in Section 2.5.1. We train the network using the L_2-constrained softmax loss described in [81]. The loss forces the features to lie on a hypersphere of a fixed radius α before applying the softmax classifier. The loss ensures that positive features are close to each other and negative features are far from one another in the cosine similarity measure. We fix the hypersphere radius to $\alpha = 50$ during training.

2.5.2.2 All-in-One-base

To ascertain that the performance improvements are not solely due to higher network complexity but due to the extra information contained in the curated dataset, we train the base network of the All-in-One CNN presented in [82]. The network consists of seven convolutional layers followed by three fully connected layers, which has a very small number of parameters compared to ResNet-101. The model was trained on the curated dataset in Section 2.5.1. At inference time, face representations are extracted from the `fc-512` layer.

2.5.2.3 Feature evaluation

To evaluate the quality of face representations learned by All-in-One-base and ResNet-101, we cluster face images in CFP and LFW using the conventional AHC algorithm. It is clear that after training on the curated dataset, both networks achieve higher performance than DNN (CASIA). Moreover, since ResNet-101 has larger capacity than All-in-One-base, it benefits even more from the additional in-



(a) CFP (b) LFW

Figure 2.6: AHC clustering performance using features extracted from DNN (CASIA), ResNet-101, and All-in-One-base.

formation provided in the curated MS-Celeb-1M dataset.

2.5.3 Quantitative Study on the CFP, LFW, and IJB-A datasets

In this section, we evaluate PAHC on CFP, LFW, and IJB-A datasets. BCubed Precision, BCubed Recall and BCubed F-measure are used as the evaluation measure. We adopt the notation in [3]. For an item e, $C(e)$ and $L(e)$ are used to denote its cluster and ground truth label, respectively. For a pair of items e and e', the relation Correct(e, e') is defined as:

$$\text{Correct}(e, e') = \begin{cases} 1, & \text{if } C(e) = C(e') \text{ and } L(e) = L(e'), \\ 0, & \text{otherwise.} \end{cases}$$

The BCubed Precision and BCubed Recall are defined as:

$$\text{Precision} = \text{Avg}_e[\text{Avg}_{e':C(e')=C(e)}[\text{Correct}(e, e')]], \tag{2.10}$$

$$\text{Recall} = \text{Avg}_e[\text{Avg}_{e':L(e')=L(e)}[\text{Correct}(e, e')]]. \tag{2.11}$$

The F-measure is the harmonic mean of the two measures, which is given by:

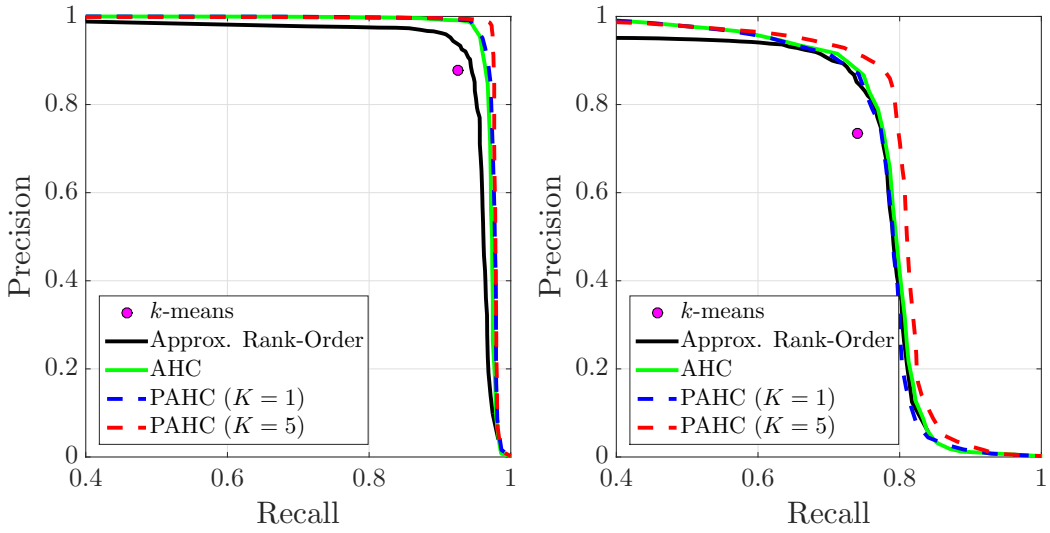
$$\text{F-measure} = \frac{2 \times \text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}}. \tag{2.12}$$

The proposed method is compared with conventional agglomerative hierarchical clustering (AHC), k-means, and Approximate Rank-Order clustering [73]. On LFW and IJB-B, we perform additional comparisons with the ConPaC algorithm proposed in [94]. In all experiments, 'Approximate Rank-Order clustering' refers to our implementation of the distance measure proposed in [73] followed by an AHC. We use the standard MATLAB implementation for hierarchical clustering and k-means, where we choose k as the true number of identities. The k-means algorithm runs with maximum number of iterations equals to 100. The best performance is reported using **bold red** and the second best is reported using **bold blue**. We use PAHC with $(K, K') = (1, 200)$, and $(K, K') = (5, 200)$ for different sizes of positive sets. As presented in Section 2.4.3, the parameter K controls the size of local neighborhoods and K' controls the amount of data used for negative samples. From our experiments, we found out that the performance of PAHC is more sensitive to K

than K'. Therefore, we fix $K' = 200$ and select $K \in \{1, 5\}$.

2.5.3.1 Analyze Feature Quality

Evaluations in Section 2.5.2 show that features extracted from the ResNet-101 have better quality than from the All-in-One-base since ResNet-101 has a higher capacity to learn from the curated MS-Celeb-1M. In Figures 2.7, 2.8, and 2.9, it is clear that by using features from ResNet-101, improved performance can be achieved for all the algorithms. Note that PAHC outperforms other methods no matter ResNet-101 or All-in-One-base features are used.

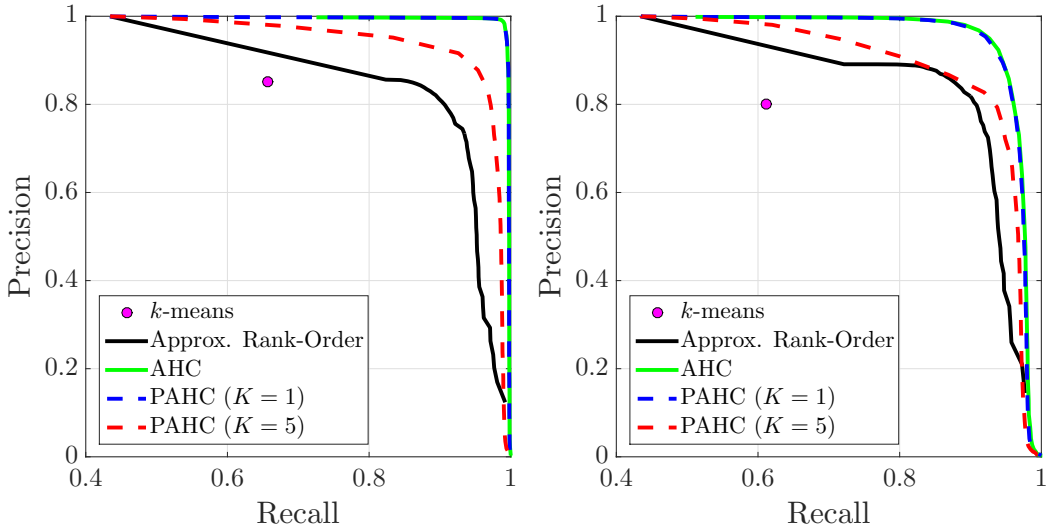


(a) ResNet-101 (b) All-in-One-base

Figure 2.7: Precision-Recall curve evaluated on the CFP dataset.

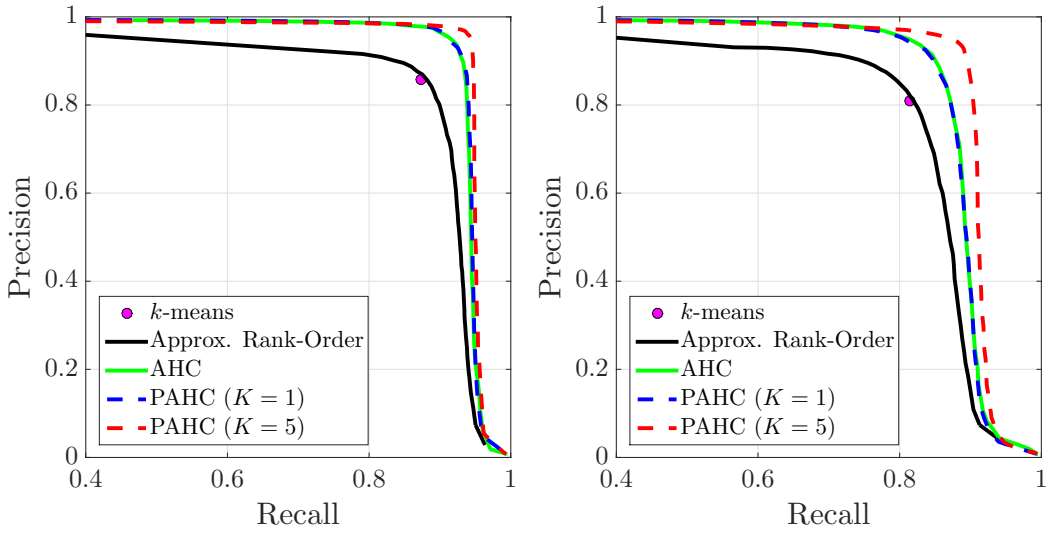
2.5.3.2 Analyze Neighborhood Size

Figures 2.7, 2.8, and 2.9 show the BCubed precision-recall performance comparisons. Table 2.1 shows optimal F-measure comparisons. It can be observed that



(a) ResNet-101 (b) All-in-One-base

Figure 2.8: Precision-Recall curve evaluated on the LFW dataset.



(a) ResNet-101 (b) All-in-One-base

Figure 2.9: Precision-Recall curve evaluated on the IJB-A dataset.

increased neighborhood size K yields improved performance on CFP and IJB-A, but degraded performance on LFW. This is because larger K generally captures more information than smaller K as on CFP and IJB-A. However, on the LFW dataset, 4,069 out of 5,749 identities in LFW have only one image. The information captured by $K = 5$ is not local anymore. Nevertheless, as shown in the figure, PAHC has the

Dataset	*CFP*		*LFW*		*IJB-A*	
Method	ResNet	AIO	ResNet	AIO	ResNet	AIO
k-means	.9005	.7379	.7422	.6930	.8657	.8117
AHC	.9643	**.8036**	**.9891**	**.9389**	.9325	**.8792**
ARO	.9332	.7975	.8577	.8646	.8731	.8251
PAHC ($K = 1$)	**.9661**	.8008	**.9894**	**.9372**	**.9336**	.8783
PAHC ($K = 5$)	**.9781**	**.8252**	.9212	.8708	**.9513**	**.9114**
ConPaC	-		.9220		-	

Table 2.1: BCubed F-measure performance evaluated on CFP, LFW, and IJB-A. The scores are reported using optimal (oracle-supplied) threshold.

Figure 2.10: One sample cluster for the CFP dataset after applying the PAHC algorithm.

flexibility of not considering neighborhood structure by setting $K = 1$. In applications such as curating large-scale datasets, it is common that multiple face images are present for each identity. The PAHC algorithm, which exploits local neighborhood structure, is then able to yield higher-quality clusters than conventional approaches by selecting some $K > 1$.

Some example clusters are shown in Figure 2.10 and Figure 2.11.

2.5.4 Quantitative Study on the IJB-B dataset

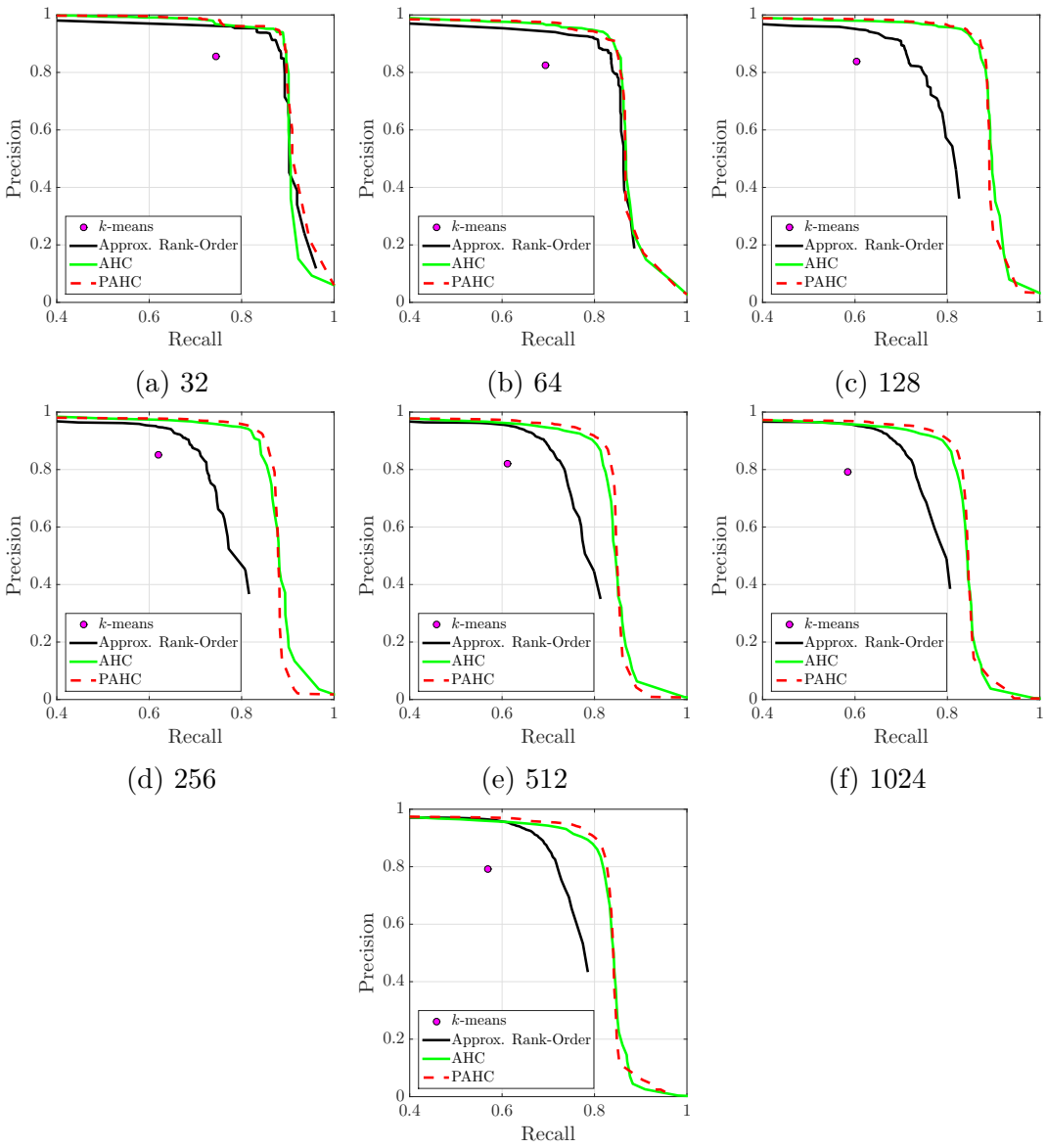
In this section, we evaluate PAHC on the IJB-B dataset using the features extracted by ResNet-101 and All-in-One-base. Based on the evaluations in Section 2.5.3, we observe that the positive set size should not be large for datasets with many singletons, as in LFW, and should be larger when there are certain neighborhood structures in the dataset, as in CFP and IJB-A. However, the IJB-B dataset

Figure 2.11: Sample clusters for the IJB-B dataset after applying the PAHC algorithm. Robustness to pose variation can be seen throughout the images. Top row shows robustness to illumination changes. Middle row shows robustness to age and makeup. Bottom row shows robustness to blur and viewpoint changes.

has large variations in cluster sizes as shown in Figure 2.5. As a result, there is no fixed positive and negative set sizes that can satisfy both singleton and non-singleton cases. To tackle this problem, we propose to associate each data point $\mathbf{x}_i$ with positive and negative sets based on the distance to its nearest neighbors. Specifically, for each $\mathbf{x}_i$, the positive set consists of half of the points with linkage cosine distance less than 0.2, and the negative set consists of 100 points with linkage cosine distance larger than 0.25. Figures 2.12 and 2.13 show BCubed precision recall curves for the seven subtasks. Table 2.2 shows optimal F-measures. The proposed method attains the best performance on several subtasks.

2.6 Conclusion

We proposed an unsupervised algorithm, namely, the PAHC algorithm, to measure the pairwise similarity between samples by exploiting their neighborhood structures. We demonstrated that the proposed algorithm has the potential to actively learn robust feature representations by harvesting information from images



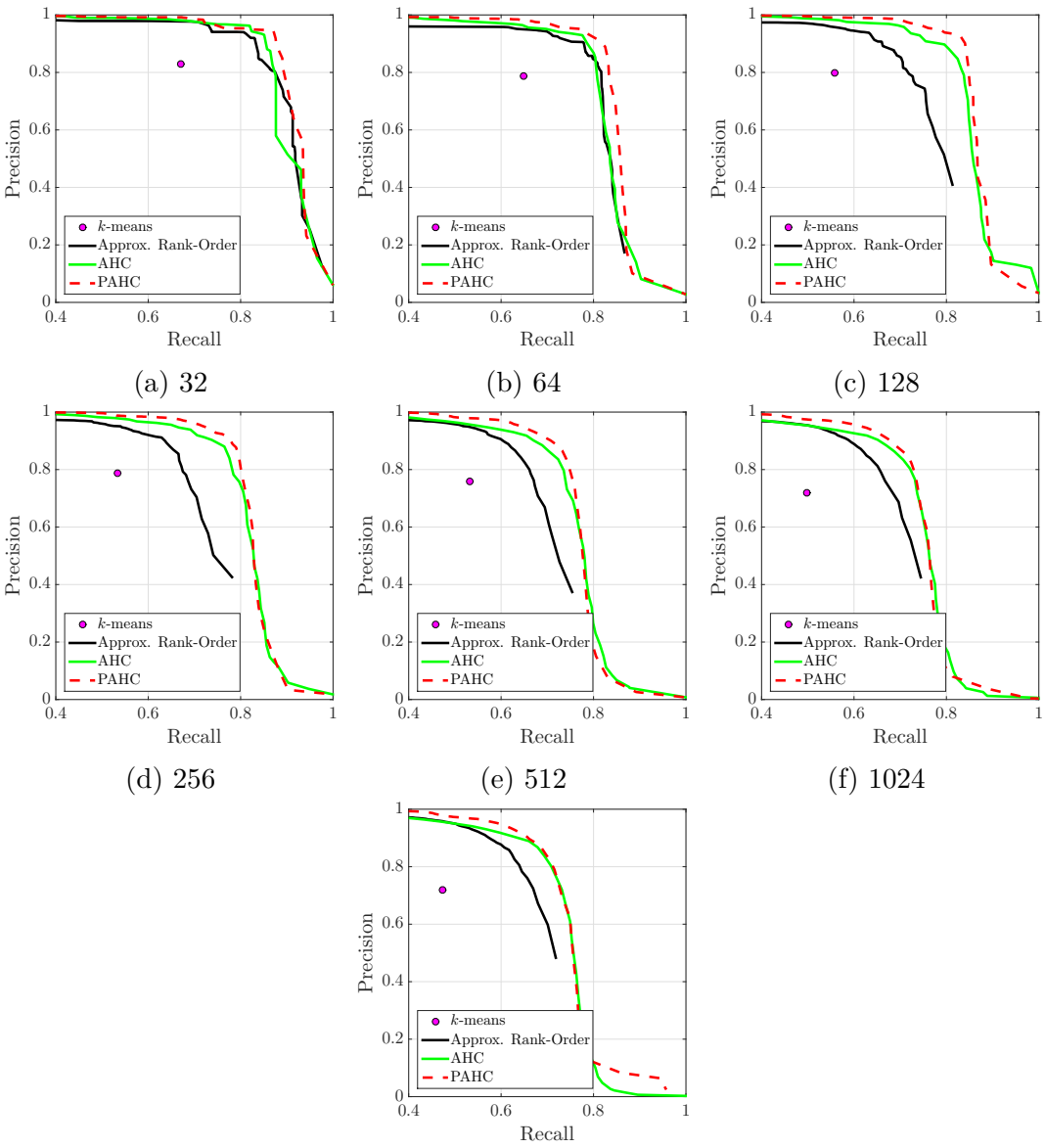
(a) 32 (b) 64 (c) 128

(d) 256 (e) 512 (f) 1024

(g) 1845

Figure 2.12: Precision-Recall curve evaluated on the IJB-B dataset using ResNet-101 features.

with noisy labels. The PAHC algorithm was first applied to curate the MS-Celeb-1M training dataset. Our algorithm retains faces with variations in pose, illumination and resolution, while separating images with different identities. We further trained two DCNNs on the curated dataset. Feature representations extracted by these networks attains extremely high performance. From extensive quantitative exper-



(a) 32 (b) 64 (c) 128

(d) 256 (e) 512 (f) 1024

(g) 1845

Figure 2.13: Precision-Recall curve evaluated on the IJB-B dataset using All-in-One-base features.

iments, we show that PAHC achieves improved precision-recall performance when the dataset has underlying local structures, which is usually the case in applications such as dataset curation.

Network	Method	Subtask						
		32	64	128	256	512	1024	1845
ResNet-101	k-means	.7958	.7539	.7020	.7167	.7011	.6730	.6623
	AHC	**.9140**	**.8762**	**.8876**	**.8738**	**.8459**	**.8419**	**.8360**
	ARO	.8967	.8576	.7911	.7794	.7830	.7799	.7765
	PAHC	**.9127**	**.8770**	**.8944**	**.8828**	**.8564**	**.8498**	**.8483**
All-in-One-base	k-means	.7409	.7105	.6567	.6363	.6256	.5885	.5706
	AHC	**.8892**	**.8458**	**.8435**	**.8184**	**.7756**	**.7652**	**.7618**
	ARO	.8728	.8365	.7703	.7483	.7296	.7231	.7180
	PAHC	**.9064**	**.8583**	**.8753**	**.8373**	**.7949**	**.7768**	**.7664**
-	ConPaC	.7510	.6560	.5630	.4930	.4810	.4520	.4290

Table 2.2: BCubed F-measure performance evaluated on the IJB-B dataset. The scores are reported using optimal (oracle-supplied) threshold.

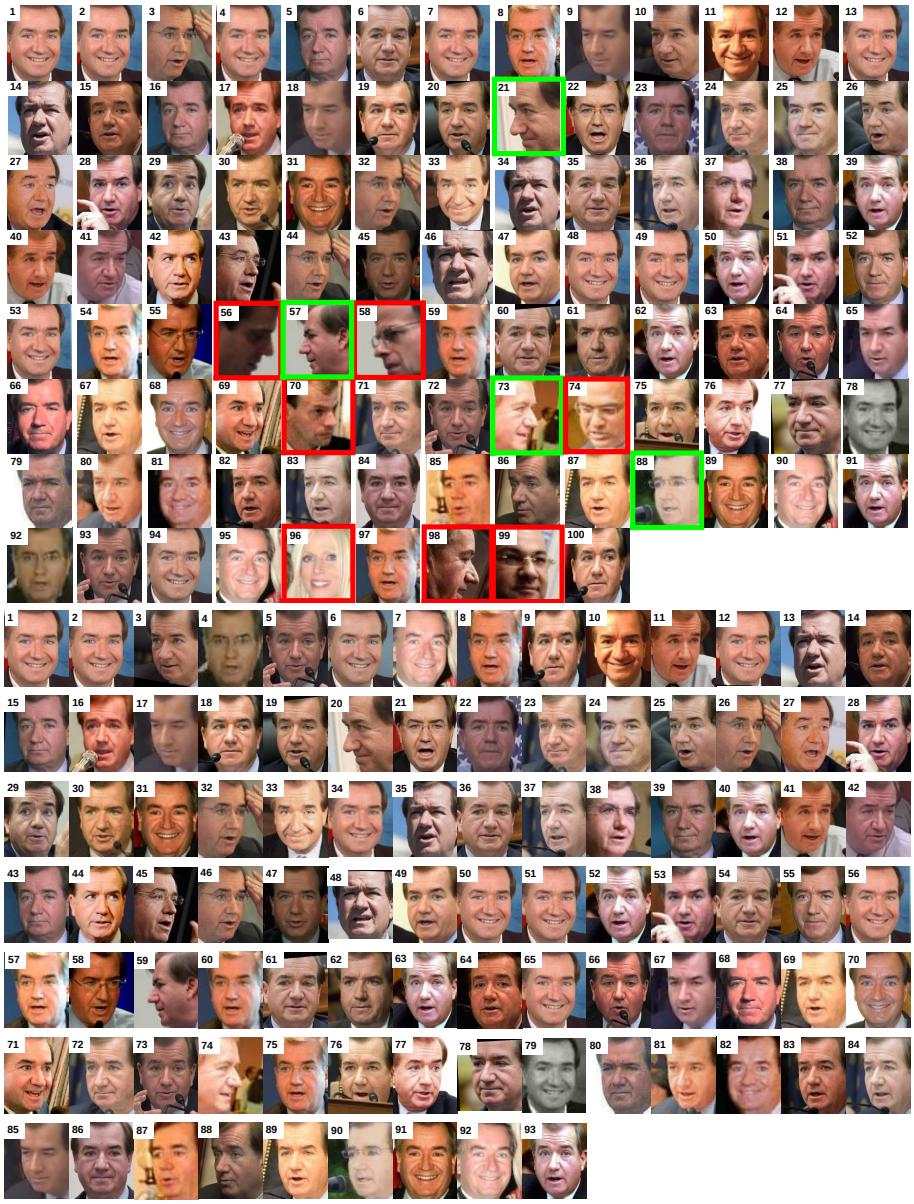


Figure 2.14: Sample face images in the MS-Celeb-1M dataset with improved purity after applying the PAHC. Upper-half of the figure shows original face images having machine identifier m.024xcy in MS-Celeb-1M dataset. The lower half of the figure is obtained following the process described in Section 2.5.1. The red boxes are the face images removed by our algorithm. The green boxes are face images that are retained by our algorithm. Variations in extreme pose (*e.g.* 21, 57, 73) and resolution (*e.g.* 88) will assist the DCNN to learn improved representation.

Chapter 3: Deep Density Clustering of Unconstrained Faces

3.1 Overview

In this chapter, we consider the problem of grouping a collection of unconstrained face images when the number of subjects is not known. We propose an unsupervised clustering algorithm called Deep Density Clustering (DDC) which is based on measuring density affinities between local neighborhoods in the feature space. By learning the minimal covering sphere for each neighborhood, information about the underlying structure is encapsulated. The encapsulation is also capable of locating high-density region of the neighborhood, which aids in measuring the neighborhood similarity. We theoretically show that the encapsulation asymptotically converges to a Parzen window density estimator. Our experiments show that DDC is a superior candidate for clustering unconstrained faces when the number of subjects is unknown. Unlike conventional linkage and density-based methods that are sensitive to the selection operating points, DDC attains more consistent and improved performance. Furthermore, the density-aware property reduces the difficulty in finding appropriate operating points.

3.2 Introduction

Given a collection of unseen face images, humans have the capability of grouping and summarizing how many distinct subjects are present by exploiting previously learned knowledge about essential components of a face and possible variations of faces from the same person. In computer vision research, this corresponds to the task of grouping visual data into clusters with targeted semantics. Most existing unsupervised algorithms group data into visually similar clusters, unaware of the underlying semantics. The success in clustering handwritten digits or faces appearing in consecutive video frames is mainly based on the fact that images that belong to the same category are visually similar. For visual data that have extreme intra-class variations, these methods may not be applicable. In this work, we focus on clustering unconstrained face images without prior knowledge of the number of distinct subjects. Visual variations caused by nuisance factors such as pose, illumination and expressions may be larger than variations between subjects. To the best of our knowledge, few previous works have addressed this challenging problem. Recent works on face clustering first extract feature vectors using deep neural networks (DNNs), and then group data directly in the feature space. Face clustering based on deep features generally has advantages over other unsupervised methods due to side information present in the training data. However, since clustering algorithms generally deal with *unseen* data, these methods will suffer from the shift in data distribution across different domains. Therefore, the underlying structure should be considered to prevent performance degradation.

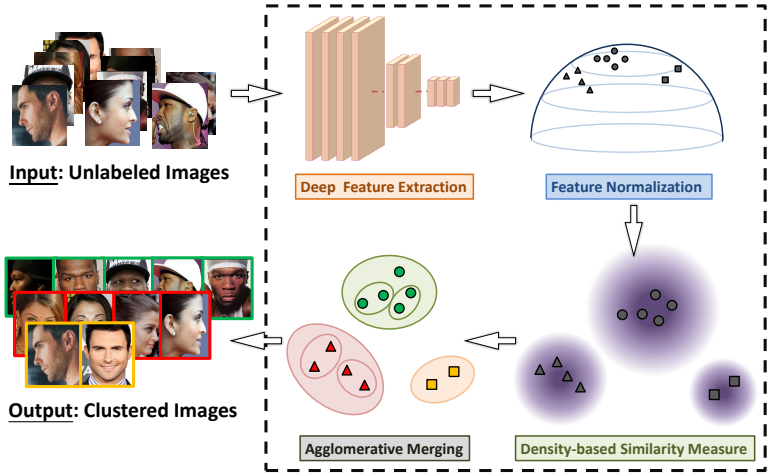


Figure 3.1: We introduce Deep Density Clustering (DDC) for unconstrained face images. DDC is a density-based clustering algorithm, which exploits the local structure of deep features for improved similarity measure.

To tackle the challenges discussed above, we propose a clustering framework, named Deep Density Clustering (DDC) that exploits the neighborhood structure of deep representations. DDC consists of three main steps: extracting deep features, computing density-based similarity, and merging clusters. The novelty is mainly in the second step: DDC first associates each data point with an ϵ-neighborhood. Points inside the neighborhood are then represented by a minimal covering sphere which encapsulates local information. Finally, DDC computes pairwise similarity by evaluating data points on the functionals defined by the spheres.

To summarize, we make the following contributions:

- A new approach for characterizing a collection of data points that encapsulates sufficient structural information in the deep feature space.

- A new method, DDC algorithm, for clustering unconstrained face images without prior knowledge of the number of subjects. We argue that information

about local structures should be included in the linkage criterion, and propose a novel similarity measure based on local density levels. We theoretically show that the similarity measure is asymptotically a Parzen window density estimator.

The remainder of this chapter is organized as follows: We first discuss several related works in unsupervised deep clustering, unconstrained face clustering, and deep representation. Then we introduce the proposed face clustering algorithm. Finally, we detail our experiments and discuss the impact of the proposed method.

3.3 Backgrounds and Related Works

In this section, we briefly introduce recent advances in unsupervised clustering and unconstrained face clustering using deep representations.

3.3.1 General Clustering Algorithms

Conventional clustering algorithms typically rely on the absolute distance defined in the embedded space. Several recent clustering algorithms, however, have shown that in addition to point-to-point topology, high-level structure could be incorporated for improved clustering performance. For example, sparse subspace clustering (SSC) [21] exploits the underlying linear subspace structure within data. Several different extensions of the SSC algorithm [75, 79, 80, 120] have yielded impressive results on MNIST [57] and Extended Yale B [26] datasets. However, the SSC algorithm relies on the assumption that the given dataset can be well-approximated

by a union of low-dimensional subspaces, which may not be true for unconstrained face images.

3.3.2 Deep Unsupervised Clustering Algorithms

Recently, deep neural networks (DNNs) are extensively used to learn representation and clusters. In [118], a recurrent framework that successively updates representations and clusters is proposed. Although good results are achieved, it requires tuning a large number of hyperparameters and repeated training of deep networks. In [27, 113, 117] encoder-decoder structures are used to learn low-dimensional embeddings and cluster assignments. Xie et al. [113] proposed to first learn deep representations using a stacked autoencoder. Cluster assignments are then iteratively refined by minimizing the KL divergence between the soft assignments and the target distribution. Yang et al. introduced a joint dimensionality reduction and clustering approach that learns a clustering-friendly latent representations. Dizaji et al. [27] proposed an end-to-end clustering framework, named DEPICT. They derived a regularized relative entropy loss function to encourage balanced clusters. In addition, the joint framework avoids layer-wise training and is computationally more efficient. Ji et al. [43] proposed the deep subspace clustering network which uses a novel self-expressive layer to mimic the self-expressiveness property. One major drawback of this method is that the number of parameters for the self-expressive layer scales quadratically with the number of images.

While successful in some applications, these methods generally require exact

knowledge of the number of categories [27, 43, 113, 117, 118], layer-wise pretraining [43, 113, 117], and tuning network structures [43, 113, 117, 118]. Furthermore, it is not clear whether clustering based on the encoder-decoder structure could be scaled to datasets with a large number of categories. In fact, the evaluations of these approaches are limited to number of clusters that are less than a hundred. The proposed DDC algorithm, on the other hand, does not require the number of categories as a prior, and is also evaluated on challenging unconstrained datasets that have more than one thousand categories.

3.3.3 Unconstrained Face Clustering

Otto et al. [73] developed an efficient algorithm called the approximated rank-order clustering that measures pairwise similarity based on the number of shared nearest neighbors. The approach of capturing the high-level structure is efficient when most of the identities have only a few instances. However, when the dataset contains more large clusters, the loss of original point-to-point topology would adversely affect the performance. Lin et al. [60] proposed the proximity-aware hierarchical clustering (PAHC) which exploits neighborhood similarity based on linear SVMs that separates local positive instances and negative instances. While improved results are achieved on unconstrained face datasets, it was applied to group faces with balanced cluster size. Unlike PAHC, the proposed method can be applied to face images with large variations in cluster sizes. Shi et al. [94] proposed the ConPaC algorithm in which the clustering problem is formulated as a conditional ran-

dom field model. By maximizing the posterior probability of the adjacency matrix, improved performance is achieved on the recently released IJB-B dataset. However, their approach does not scale well in speed. The proposed DDC algorithms, on the other hand, could run significantly faster than the ConPaC algorithm. Jin et al. [45] proposed the Erdős-Rényi clustering algorithm for joint face detection and clustering in videos. The algorithm is based on the rank-1 count similarity which requires a reference set. In their work, a collection of frames are sampled as the reference set that are likely to have similar distribution to the target distribution. However, collecting such reference set for general face clustering is not an easy task. Unlike the Erdős-Rényi clustering algorithm, our approach does not require a domain-specific reference set.

3.3.4 Deep Face Representations

Deep convolutional neural networks (DCNNs) have been widely used for face classification [63, 91, 98]. A DCNN trained on labeled face images is able to separate faces from distinct identities in the embedded feature space. In this work, we use deep face representations for *unseen* face images to retain sufficient amount of semantic information for distinguishing different identities.

3.4 Proposed Method

For an unlabeled dataset $X = [\boldsymbol{x}_1, \cdots, \boldsymbol{x}_N] \in \mathbb{R}^{D \times N}$, the goal of unsupervised clustering algorithm is to find proper cluster assignment for each data point, such

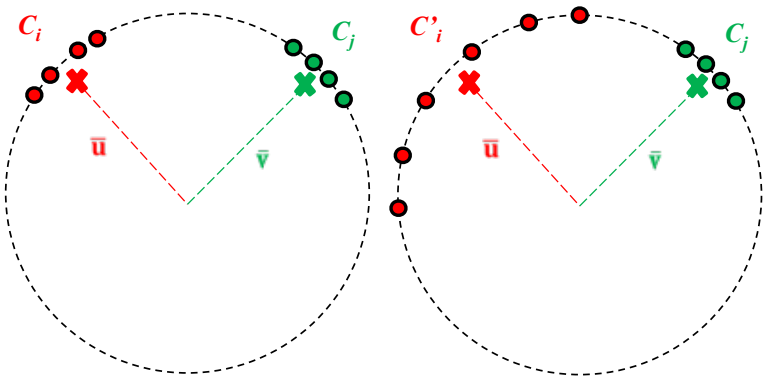


Figure 3.2: Linkage computation for two groups of data points on a circle. It is clear that after averaging, $\bar{u}$ and $\bar{v}$ fail to represent whether the original group of points are sparsely or densely distributed.

that data of the same state-of-the-nature identity are grouped together. In this work, we consider X as a collection of unconstrained face images with unknown number of subjects. We adopt the basic average-linkage clustering approach, in which pairs of face images are grouped according to (1) the distance measure in the embedded space and (2) the average linkage criterion that measures the dissimilarity between two groups of face images.

For unconstrained face images, within-subject variations could be larger than between-subject variations. To capture sufficient amount of semantic information for distinguishing different subjects, face images are first projected into the embedded space using a DNN $\Psi_\theta : \mathbb{R}^D \rightarrow \mathbb{R}^d$. Recent works [63] on deep representations have shown that for DNNs trained with softmax loss, label prediction is mainly determined by angular similarities to each class. Therefore, we consider cosine distance as the distance measure in the feature space. Without loss of generality, $\Psi_\theta : \mathbb{R}^D \rightarrow \mathbb{S}^{d-1}$ is used to represent a DNN, where $\mathbb{S}^{d-1}$ is a unit hypersphere.

3.4.1 Key Observations

In this section, we first show that point-to-point distance measurement might be insufficient, and then describe the motivation for the proposed method.

In Figure 3.2, the average linkage between two groups of points C_i and C_j determines whether they should be merged together. By definition, if the distance measure is d, the average linkage is calculated by

$$d(C_i, C_j) = \frac{1}{|C_i||C_j|} \sum_{\boldsymbol{u} \in C_i, \boldsymbol{v} \in C_j} d(\boldsymbol{u}, \boldsymbol{v}). \tag{3.1}$$

For data points that lie on a unit hypersphere S^{d-1}, (3.1) equals to $1 - \bar{\boldsymbol{u}}^T \bar{\boldsymbol{v}}$, which is equivalent to the cosine distance between arithmetic averages $\bar{\boldsymbol{u}}$ and $\bar{\boldsymbol{v}}$. Note that local information about C_i is not retained in $\bar{\boldsymbol{u}}$ and $\bar{\boldsymbol{v}}$. One can find another sparsely distributed C_i' with the same $\bar{\boldsymbol{u}}$. However, merging C_i' and C_j is less desirable since the cluster $C_i' \cup C_j$ is less homogeneous than $C_i \cup C_j$. We argue that neighborhood information should be aggregated during linkage computation in order to differentiate merging C_i or C_i' with C_j. Specifically, when measuring the distance between two points, their neighboring points should also be considered. Following this observation, we propose a new similarity measure based on the following steps: (1) building a nearest-neighbor graph for the entire dataset, which will be described in Section 3.4.2, (2) representing each neighborhood in a compact form, which will be discussed in Section 3.4.3, and (3) computing a density-based similarity, which will be described in Section 3.4.4. We name the proposed method Deep Density

Clustering since the similarity measure is asymptotically a Parzen window density estimator as will be proved in Section 3.4.4.

3.4.2 Nearest-Neighbor Graph Construction

We can view a set of data points as a union of local neighborhoods. Namely, we can write $X = \bigcup_{m=1}^{N} V(\boldsymbol{x}_m)$, where $V(\boldsymbol{x}_m)$ consists of neighboring points of $\boldsymbol{x}_m$ measured in the feature space. Common approaches to constructing local neighborhoods include k-nearest neighbor rule and ϵ-neighborhood criterion. We construct $V(\boldsymbol{x}_m)$ based on the ϵ-neighborhood approach since it is more robust to density variations, and as $N \to \infty$, $|V(\boldsymbol{x})| \to \infty$ holds, which achieves the asymptotic property that will be discussed in subsequent sections. However, proper selection of ϵ is not trivial and usually depends on the representation. In this work, we propose to select ϵ as the maximum likelihood (ML) estimator of the cosine distance between *matched pairs* (image pairs belong to the same subjects). Formally,

$$\epsilon = \underset{d}{\operatorname{argmax}}\, \mathbb{E}_c[p(d \mid c)], \tag{3.2}$$

where c is the subject label. The matched pairs can be sampled from the training data or an external dataset. Details about the selection of ϵ will be presented in Section 3.5.1.

3.4.3 Local Neighborhood Encapsulation

A trivial way of characterizing points in a neighborhood is to store all the points, however, this representation will not be useful. We propose to encapsulate each local neighborhood in a hypersphere which retains information about local structure. This is inspired by the SVDD algorithm [99] that describes a collection of data by finding a sphere that covers all the target data while including no superfluous space. Instead of the entire dataset, we apply SVDD to all the local neighborhoods. For each $V(x_m)$, we solve for its encapsulation using the following optimization:

$$\begin{aligned} \min_{\boldsymbol{c}_m, \bar{R}_m, \boldsymbol{\xi}_m} \quad & \bar{R}_m + \frac{1}{\nu \cdot n_V} \sum_{\boldsymbol{z} \in V(\boldsymbol{x}_m)} \xi_m(\boldsymbol{z}) \\ \text{s.t.} \quad & \|\Psi_\theta(\boldsymbol{z}) - \boldsymbol{c}_m\|^2 \leq \bar{R}_m + \xi_m(\boldsymbol{z}),\ \xi_m \geq 0,\ \forall \boldsymbol{z} \in V(\boldsymbol{x}_m), \end{aligned} \tag{3.3}$$

where $\bar{R}_m = R_m^2$ is the squared radius and n_V is the size of $V(\boldsymbol{x}_m)$. Note that in (3.3), instead of minimizing over R_m, we aim to solve for optimal $\bar{R}_m^*$ since the original formulation in [99] is not convex. Readers are referred to [12] for more details. After solving (3.3) for $m = 1, \cdots, N$, the resulting collection of spheres $\{(\boldsymbol{c}_m^*, R_m^*)\}_{m=1}^N$ minimally covers each local neighborhood as demonstrated in Figure 3.3. In what follows, when there is no confusion possible, we will drop the subscript m in (3.3) for more compact notations.

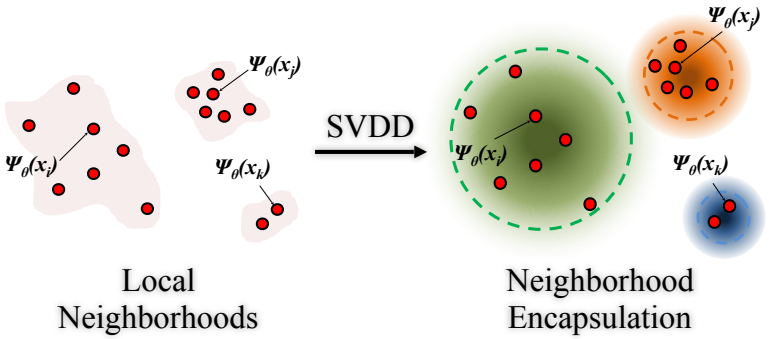


Figure 3.3: Neighborhood encapsulation. (left) Pink regions are the local neighborhoods of the points x_i, x_j, and x_k in feature space. (right) Encapsulations are learned by solving (3.3). The encapsulation is density-aware. In the figure, regions closer to the centers of the spheres have higher density.

3.4.3.1 Relation to One-Class SVM

One-class SVM (OC-SVM) was first proposed in [90] to build a representational model for a given dataset. Suppose we choose the set as $V(\boldsymbol{x})$, then OC-SVM aims to solve the following optimization problem:

$$\begin{aligned} \min_{\boldsymbol{w},\,\rho,\,\boldsymbol{\xi}} \quad & \frac{1}{2}\|\boldsymbol{w}\|^2 + \frac{1}{\nu \cdot n_V}\sum_{\boldsymbol{z}\in V(\boldsymbol{x})}\xi_{\boldsymbol{z}} - \rho \\ \text{s.t.} \quad & \boldsymbol{w}^T\Psi_\theta(\boldsymbol{z}) \geq \rho - \xi_{\boldsymbol{z}}, \\ & \xi_{\boldsymbol{z}} \geq 0, \quad \forall \boldsymbol{z} \in V(\boldsymbol{x}). \end{aligned} \tag{3.4}$$

The optimal hyperplane separates the data with the origin in feature space and maximizes the distance from the hyperplane to the origin. We present the following Lemma showing equivalence between the formulations in (3.3) and (3.4).

Lemma 1 *If $1/n_V < \nu \leq 1$, the SVDD formulation in (3.3) is equivalent to the OC-SVM formulation in (3.4) when the evaluation functions for the two are given*

by

$$h_{SVDD}(\boldsymbol{x}) = \bar{R}^* - \left\|\Psi_\theta(\boldsymbol{x}) - \boldsymbol{c}^*\right\|^2, \tag{3.5}$$

$$h_{OC\text{-}SVM}(\boldsymbol{x}) = \boldsymbol{w}^{*T}\Psi_\theta(\boldsymbol{x}) - \rho^*, \tag{3.6}$$

with the correspondence $\boldsymbol{w}^* = \boldsymbol{c}^*$*, and* $\rho^* = \boldsymbol{c}^{*T}\Psi_\theta(\boldsymbol{x}_s)$*, where* $\boldsymbol{x}_s$ *is a support vector in* (3.3) *that lies on the learned enclosing sphere.*

Proof. The condition corresponds to the case $1/n_V \leq C < 1$ in [12] with $C = 1/(\nu \cdot n_V)$. We introduce the kernel function $K(\boldsymbol{x}_i, \boldsymbol{x}_j) = \Psi_\theta(\boldsymbol{x}_i)^T\Psi_\theta(\boldsymbol{x}_j)$. Since $K(\boldsymbol{x}_i, \boldsymbol{x}_i)$ is constant in our setting, the same dual formulation for (3.3) and (3.4) can be written as:

$$\min_{\boldsymbol{\alpha}} \sum_{ij} \alpha_i \alpha_j K(\boldsymbol{x}_i, \boldsymbol{x}_j) \quad \text{s.t.} \quad 0 \leq \alpha_i \leq C, \ \sum_{i=1}^{n_V} \alpha_i = 1.$$

Let $S = \{i \mid 0 < \alpha_i < C\}$. We have the following results:

$$\boldsymbol{c}^* = \sum_{i=1}^{n_V} \alpha_i \Psi_\theta(\boldsymbol{x}_i), \quad \bar{R}^* = \left\|\Psi_\theta(\boldsymbol{x}_s) - \boldsymbol{c}^*\right\|^2, \tag{3.7}$$

$$\boldsymbol{w}^* = \sum_{i=1}^{n_V} \alpha_i \Psi_\theta(\boldsymbol{x}_i), \quad \rho^* = \boldsymbol{w}^{*T}\Psi_\theta(\boldsymbol{x}_s), \tag{3.8}$$

where $s \in S$. Substituting into (3.5) and (3.6), we obtain

$$h_{SVDD}(\boldsymbol{x}) = 2 \cdot h_{OC\text{-}SVM}(\boldsymbol{x}) = 2\left[\sum_{i=1}^{n_V} \alpha_i K(\boldsymbol{x_i}, \boldsymbol{x}) - \rho^*\right]. \tag{3.9}$$

Intuitively, the evaluation functions (3.5) and (3.6) measures the closeness to the neighborhood $V(\boldsymbol{x})$.

3.4.4 Density-based Similarity Measure

Our goal is to associate each pair of points with a similarity measure. We first use the following theorem to show the evaluation function defined in (3.5) is a local density estimator.

Theorem 3.1 *If* $1/n_V < \nu \leq 1$ *and* $\boldsymbol{c}^{*T}\Psi_\theta(\boldsymbol{x}_s) \neq 0$ *for some support vector* $\boldsymbol{x}_s$, $h_{SVDD}(\boldsymbol{x})$ *defined in* (3.5) *is asymptotically a Parzen window density estimator in the feature space with Epanechnikov kernel.*

Proof. Given the condition, according to Lemma 1, $h_{SVDD}(\boldsymbol{x})$ is equivalent to $h_{OC\text{-}SVM}(\boldsymbol{x})$ with $\rho^* \neq 0$. From the results in [88] and the fact that $\sum \alpha_i = 1$ in the dual formulations of (3.3) and (3.4), it can be shown that

$$h_{OC\text{-}SVM}(\boldsymbol{x}) = \frac{8}{3}\sum_{i=1}^{n_V} \alpha_i K_E\left(\frac{\|\Psi_\theta(\boldsymbol{x}) - \Psi_\theta(\boldsymbol{x}_i)\|}{2}\right) - \rho^* - 1,$$

where $K_E(u) = \frac{3}{4}(1-u^2),\ |u| \leq 1$ is the Epanechnikov kernel. As a consequence of Proposition 4 in [88] and the proof of Proposition 1 in [89], when $n_V \to \infty$, the fraction of support vector is ν, and the fraction of points with $0 < \alpha_i < 1/(\nu \cdot n_V)$ vanishes. Therefore, either $\alpha_i = 0$ or $\alpha_i = 1/(\nu \cdot n_V)$. By introducing the notation

$\bar{S} = \{i \mid \alpha_i = 1/(\nu \cdot n_V)\}$, it can be shown that

$$h_{OC\text{-}SVM}(\boldsymbol{x}) = \frac{2^{d+3}}{3} \hat{f}\left(\Psi_\theta(\boldsymbol{x})\right) - \rho^* - 1, \tag{3.10}$$

where $\hat{f}(\boldsymbol{z}) = \frac{1}{\nu \cdot n_V \cdot 2^d} \sum_{s \in \bar{S}} K_E\left(\frac{\|\boldsymbol{z}_s - \boldsymbol{z}\|}{2}\right)$ is a density estimator. As a result, $h_{SVDD}(\boldsymbol{x})$ is equivalent to a Parzen window density estimator with Epanechnikov kernel of bandwidth 2. By scaling properly, Parzen window estimator with different bandwidths can be obtained.

According to Theorem 3.1, we associate each neighborhood $V(\boldsymbol{x}_m)$ with a density estimator $\mathcal{E}_m : \mathbb{R}^D \to \mathbb{R}$:

$$\mathcal{E}_m(\boldsymbol{x}) = \bar{R}^*_m - \left\|\Psi_\theta(\boldsymbol{x}) - \boldsymbol{c}^*_m\right\|^2. \tag{3.11}$$

Data points that yield smaller $\mathcal{E}_m$ lie in low-density region of $V(\boldsymbol{x}_m)$ and are therefore less similar to the neighborhood $V(\boldsymbol{x}_m)$. This leads to a similarity measure between $\boldsymbol{x}_i$ and $\boldsymbol{x}_j$, which is defined as:

$$s(\boldsymbol{x}_i, \boldsymbol{x}_j) = \frac{1}{2}\left[\frac{\sum_{\boldsymbol{z} \in V(\boldsymbol{x}_j)} \mathcal{E}_i(\boldsymbol{z})}{|V(\boldsymbol{x}_j)|} + \frac{\sum_{z \in V(\boldsymbol{x}_i)} \mathcal{E}_j(\boldsymbol{z})}{|V(\boldsymbol{x}_i)|}\right]. \tag{3.12}$$

The distance between pairs of data samples can be taken as a proper monotonically decreasing function of $s(\boldsymbol{x}_i, \boldsymbol{x}_j)$.

3.4.5 Negative Set Mining

In [100], the authors proposed that when negative samples are available for SVDD, they can be incorporated to improve the description. Specifically, the enclosing sphere is refined by modifying the constraints in the following way:

$$\|\Psi_\theta(\boldsymbol{z}) - \boldsymbol{c}\|^2 \leq \bar{R} + \xi, \quad \forall \boldsymbol{z} \in V(\boldsymbol{x}), \tag{3.13}$$

$$\|\Psi_\theta(\boldsymbol{z}) - \boldsymbol{c}\|^2 \geq \bar{R} - \xi, \quad \forall \boldsymbol{z} \in V^-(\boldsymbol{x}), \tag{3.14}$$

where $V^-(\boldsymbol{x}) \subset X$ contains instances which are *hard negatives* of $\boldsymbol{x}$. However, since no label information about $\boldsymbol{x}$ is available, the general notion of negative samples is not well-defined. Instead, we view the selection of hard negative samples as finding a balance between the amount of false positives and the false negatives in binary hypothesis testing formulation. Specifically, points in $V^-(\boldsymbol{x})$ are sampled from $\{\boldsymbol{x}' : d(\boldsymbol{x}', \boldsymbol{x}) > \eta\}$, where η is chosen to minimize the misclassification rate. In other words, we assign the same risk function to the action of selecting false positives and false negatives. Details about the selection of η will be presented in Section 3.5.1.

To incorporate the negative samples, we make the following observations. From the equivalence of (3.3) and (3.4), when no negative samples are available, encapsulations are learned by separating the data points against the origin with a margin ρ. Note that the $-\rho$ in (3.4) encourages large separation with the origin. In the presence of negative samples, $-\rho$ is no longer required, and the hyperplane

in (3.4) should target at separating positive and negative samples. We propose to learn the set of enclosing spheres such that positive and negative examples are separated by a margin Δ. Therefore, by the equivalence of (3.5) and (3.6) and the arguments given above, we formulate the algorithm with negative set mining as:

$$\begin{aligned} \min_{\boldsymbol{w},\rho,\boldsymbol{\xi}} \quad & \frac{1}{2}\|\boldsymbol{w}\|^2 + C\sum \xi_z \\ \text{s.t.} \quad & \boldsymbol{w}^T\Psi_\theta(\boldsymbol{z}) - \rho \geq \Delta - \xi_z, \quad \forall \boldsymbol{z} \in V(\boldsymbol{x}), \\ & \boldsymbol{w}^T\Psi_\theta(\boldsymbol{z}) - \rho \leq -\Delta + \xi_z, \forall \boldsymbol{z} \in V^-(\boldsymbol{x}), \\ & \xi_z \geq 0, \qquad \forall \boldsymbol{z}. \end{aligned} \tag{3.15}$$

Note that Δ is not a hyperparameter since we can divide both sides of the constraints by Δ and obtain a large margin formulation with L_1 normalization.

3.5 Evaluation and Discussion

In this section, we evaluate the proposed clustering approach on YouTube Faces Database (YTF), Labeled Faces in the Wild (LFW) and IARPA JANUS Benchmark B (IJB-B) datasets. The datasets are briefly described as follows:

- **YouTube Faces Database (YTF) [109]:** The dataset contains 3,425 videos of 1,595 different people. We choose the first 41 subjects from the YTF dataset as in [27, 118].

- **Labeled Faces in the Wild (LFW) [38]:** It is a well-known and standard dataset for unconstrained face recognition which contains 13,233 images of

(a) YTF

(b) LFW

(c) IJB-B

Figure 3.4: Sample images for the datasets.

5749 subjects. Note that 4169 subjects of the dataset have only one image. We evaluate the proposed approach using the entire dataset.

- **IARPA JANUS Benchmark B (IJB-B) [108]:** The IJB-B dataset contains 1,845 subjects with 11,754 images, 55026 video frames and 10,044 non-face images. It contains a clustering protocol, which consists of seven subtasks. These subtasks differ in the number of distinct identities and the number of face images. Many face images are in extreme poses or of low quality, making the dataset more challenging than YTF and LFW. We evaluate clustering algorithms on four subtasks with number of identities 128, 256, 1024 and 1845.

Dataset	# Samples	# Subjects
YTF	10,000	41
LFW	13,233	5,749
IJB-B-128	5,224	128
IJB-B-256	9,867	256
IJB-B-1024	36,575	1,024
IJB-B-1845	68,195	1,845

Table 3.1: Datasets used in the experiments.

3.5.1 Implementation Details

Deep Face Representation. We adopt the network architecture presented in [130]. The network is first trained on the CASIA-WebFace dataset [119] using SGD for 750K iterations with a standard batch size 128 and momentum 0.9. Then, the model is finetuned for 230K iterations using the MSCeleb-1M dataset [34]. The inputs to the networks are 100×100×3 RGB images. Data augmentation is performed by randomly cropping and horizontally flipping face images. Given a face image, the deep representation is extracted from the `pool5` layer with dimension 320.

Parameter Selection. There are two main hyperparameters in the proposed approach: ϵ for constructing neighborhoods and η for mining hard negatives. To select ϵ, we follow (3.2) by randomly sampling 100 subjects from the training dataset and computing cosine distance between all matched pairs. The red curve in Figure 3.5 represents the fitted distribution. The ML estimate is therefore $\epsilon \approx 0.23$. The green curve in Figure 3.5 represents the distribution of the cosine distance between mismatched pairs among the 100 subjects. From Figure 3.5, it is clear that $\eta \approx 0.40$ minimizes the Bayesian risk of selecting false positives and false negatives.

We use the default parameters provided with the code[1] when solving (3.3) or (3.15).

[1] `https://www.csie.ntu.edu.tw/~cjlin/libsvmtools/`

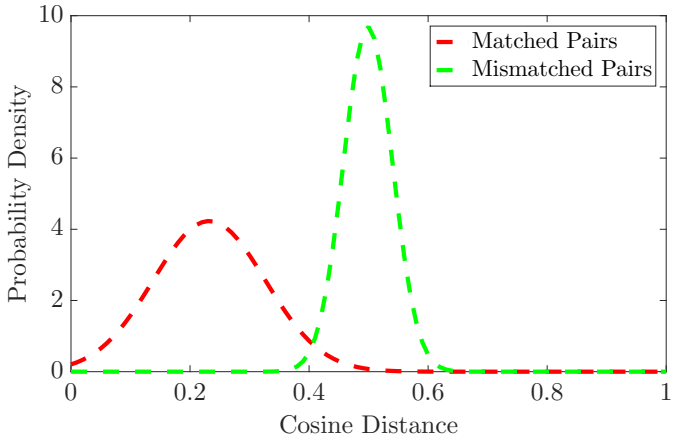


Figure 3.5: Distribution of cosine distance from the training dataset.

3.5.2 Evaluation Metrics

To evaluate clustering algorithms, we adopt two measures: normalized mutual information (NMI) and BCubed F-measure [3].

NMI is a widely used metric that measures the normalized similarity between the ground truth labels and the labels decided by the clustering algorithms. NMI is suitable for evaluation when the number of clusters is assumed to be a known quantity. However, when the number of clusters is unknown or is the quantity we are trying to estimate, NMI may fail to penalize algorithms that yield over-clusterings. We use NMI mainly for comparing with other state-of-the-art unsupervised image clustering methods.

BCubed F-measure [3] is the harmonic mean of BCubed precision and BCubed recall. BCubed precision calculates the fraction of points in the same cluster that belong to the same class. BCubed recall calculates the fraction of points in the same class that are assigned to the same cluster. Formally, for an item e, $C(e)$ and $L(e)$

are used to denote its cluster and ground truth label, respectively. For a pair of items e and e', the relation $\text{Correct}(e, e')$ is defined as:

$$\text{Correct}(e, e') = \begin{cases} 1, & \text{if } C(e) = C(e') \text{ and } L(e) = L(e'), \\ 0, & \text{otherwise.} \end{cases} \tag{3.16}$$

The BCubed Precision, BCubed Recall, and BCubed F-measure are defined as:

$$\text{Precision} = \text{Avg}_e[\text{Avg}_{e':C(e')=C(e)}[\text{Correct}(e, e')]], \tag{3.17}$$

$$\text{Recall} = \text{Avg}_e[\text{Avg}_{e':L(e')=L(e)}[\text{Correct}(e, e')]]. \tag{3.18}$$

$$\text{F-measure} = \frac{2 \times \text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}}. \tag{3.19}$$

BCubed Precision and Recall can be used to evaluate clustering algorithms that yield different number of clusters. They satisfy several formal constraints on evaluation metrics, and is shown to be more suitable than metrics based on set matching, pair counting, entropy or editing distance [3].

3.5.3 Baseline Methods

We compare the proposed DDC algorithm, DDC with negative set mining (DDC-NEG), with the following methods: Agglomerative Hierarchical Clustering

Methods		DDC-NEG	DDC	PAHC	DBSCAN	AHC
YTF	0.70	1.000	1.000	0.168	1.000	1.000
	0.75	1.000	1.000	0.144	1.000	1.000
	0.80	0.999	1.000	0.120	1.000	0.999
	0.85	0.958	0.966	0.098	0.990	0.948
LFW	0.70	0.994	0.992	0.976	1.000	1.000
	0.75	0.991	0.991	0.956	0.995	1.000
	0.80	0.991	0.991	0.919	0.936	0.994
	0.85	0.990	0.990	0.822	0.664	0.990
IJB-B-128	0.70	**0.966**	**0.960**	0.431	0.842	0.947
	0.75	**0.913**	**0.857**	0.253	0.705	**0.913**
	0.80	**0.786**	0.504	0.172	0.461	**0.679**
	0.85	**0.411**	0.225	0.156	**0.275**	0.253
IJB-B-256	0.70	**0.937**	0.901	0.169	0.725	**0.915**
	0.75	**0.893**	0.760	0.132	0.592	**0.868**
	0.80	**0.620**	0.396	0.102	0.395	**0.524**
	0.85	**0.181**	0.126	0.079	**0.230**	0.139
IJB-B-1024	0.70	**0.798**	0.616	0.087	0.485	**0.735**
	0.75	**0.459**	0.210	0.053	**0.347**	0.307
	0.80	**0.105**	0.101	0.038	**0.241**	0.055
	0.85	0.050	**0.066**	0.022	**0.157**	0.025
IJB-B-1845	0.70	**0.771**	0.610	0.059	0.492	**0.690**
	0.75	**0.341**	0.204	0.045	**0.350**	0.235
	0.80	**0.083**	0.081	0.031	**0.233**	0.052
	0.85	**0.068**	0.051	0.018	**0.151**	0.019

Table 3.2: BCubed precision evaluated at different BCubed recall values. The best performance is reported using **bold red**, and the second best is reported using **bold blue**.

(AHC) [31], K-means [64], Density-Based Spatial Clustering of Applications with Noise (DBSCAN) [22], Affinity Propagation (AP) [24], Sparse Subspace Clustering using Orthogonal Matching Pursuit (SSC-OMP) [120], Joint Unsupervised Learning of deep representations and clusters (JULE) [118], Deep Embedded Regularized Clustering (DEPICT) [27], Proximity-Aware Hierarchical Clustering (PAHC) [60], Approximate Rank-Order Clustering (ARO) [73], and Conditional Pairwise Clustering (ConPaC) [94].

Precision and Recall Comparisons. Table 3.2 shows the BCubed precision measured at different BCubed recall for methods that yields different number of clusters. On the YTF and LFW datasets, all methods except PAHC and DBSCAN attains near-perfect performance. On the more-challenging IJB-B dataset, the proposed approach performs the best across several subtasks. It should be noted that

the proposed approach has consistent behavior across different operating points and dataset scales, while the basic AHC achieves degraded performance at higher recall regions for larger scale data, and DBSCAN is inferior at lower recall regions.

F-measure and NMI Comparisons. Table 3.4 reports the F-measure and NMI comparisons. Some experiments for SSC-OMP do not finish within the cut-off threshold of ten hours and are replaced by double dash marks (- -). Results reported from the original papers are marked by asterisks (*). As shown in the table, the proposed DDC and DDC-NEG outperforms other methods. Although AHC achieves high F-measure and NMI using the oracle supplied threshold, it is inferior at other operating points as discussed in the previous section.

Note that we view JULE, DEPICT and DDC as solving a complete different problem. Given a collection of unseen face images, it is not practical to assume the number of subjects to be a known quantity. Furthermore, the number of classes reflects the complexity of the data at hand. Without this information, methods such as JULE and DEPICT may suffer from tuning network structures. Therefore, the proposed algorithm is more suitable for applications in which the number of clusters is not known.

Discussion. We observe from the statistics in Table 3.1 that LFW contains a large number of singleton clusters and YTF consists of multiple large clusters. Since the AHC algorithm uses cosine similarity as the underlying measure, in LFW, it exploits the discriminative power of deep features in 1-1 comparisons (verification) and hence high performance is achieved. However, AHC exhibits inferior performance in YTF, since it ignores local structures as presented in Section 3.4.1.

Both DBSCAN and PAHC are aware of local neighborhoods with fixed sizes. DBSCAN attains improved performance for larger clusters, and PAHC performs well on template-based data [60]. However, since the neighborhood sizes are not adaptive to local density variations, DBSCAN has degraded performance on LFW, and PAHC does not achieve comparable performance with other methods. The proposed algorithm attains improved performance by balancing discriminative power and density-aware property.

Running Time Comparisons. We compare the running time performance using IJB-B-1024 and IJB-B-1845 subtasks which contain 36,575 and 68,195 faces respectively. The results are reported in Table 3.3.

Dataset	*IJB-B-1024*	*IJB-B-1845*
K-means [64]	00:00:17	00:01:00
AHC [31]	00:00:29	00:01:32
DBSCAN [22]	00:07:49	00:49:31
AP [24]	03:55:42	08:42:50
PAHC [60]	00:01:19	00:03:56
ARO [73]	00:00:37	00:00:73
ConPaC [94]	00:20:06	02:53:58
DDC	00:02:17	00:05:32
DDC-NEG	00:01:55	00:05:39

Table 3.3: Running Time Comparisons (HH:MM:SS).

3.5.4 Determining Operating Point

The reported performance on different operating points is obtained by thresholding the pairwise similarity matrix at different levels: large thresholds result in several tiny clusters which correspond to high precision and low recall operating points, while small thresholds result in a few gigantic clusters which correspond

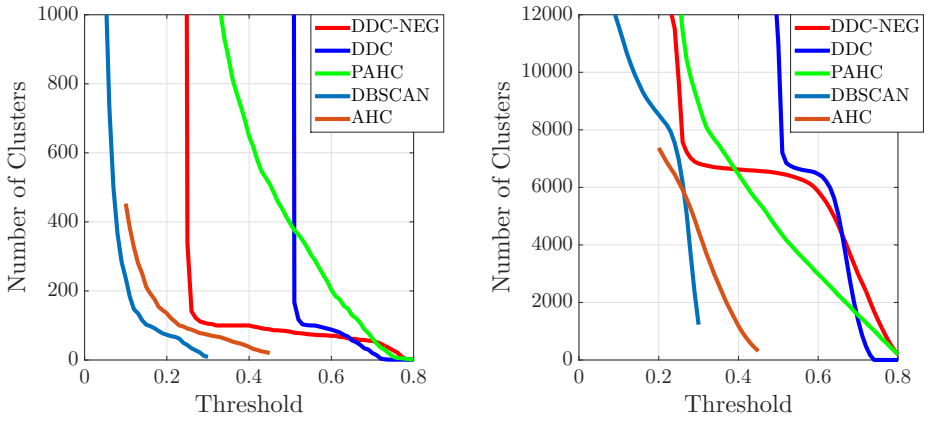
Dataset	*YTF*		*LFW*		*IJB-B-128*		*IJB-B-256*		*IJB-B-1024*		*IJB-B-1845*	
	F	NMI	F	NMI	F	NMI	F	NMI	F	NMI	F	NMI
K-means [64]	0.815	0.915	0.688	0.922	0.628	0.835	0.585	0.838	0.551	0.851	0.532	0.854
AHC [31]	0.908	0.960	0.940	0.987	0.824	0.925	0.805	0.922	0.736	0.919	0.729	0.921
AP [24]	0.312	0.795	0.618	0.906	0.439	0.822	0.426	0.836	0.411	0.854	0.405	0.858
DBSCAN [22]	**0.923**	**0.967**	0.868	0.973	0.777	0.893	0.762	0.895	0.675	0.894	0.672	0.895
SSC-OMP [120]	0.142	0.174	- -	- -	0.177	0.476	0.136	0.483	- -	- -	- -	- -
JULE* [118]	-	0.848	-	-	-	-	-	-	-	-	-	-
DEPICT* [27]	-	0.802	-	-	-	-	-	-	-	-	-	-
PAHC [60]	0.360	0.734	0.857	0.958	0.695	0.863	0.648	0.865	0.639	0.890	0.610	0.890
ARO* [73]	-	-	0.870	-	0.482	-	0.423	-	0.352	-	0.317	-
ConPaC* [94]	-	-	0.922	-	0.563	-	0.493	-	0.452	-	0.429	-
DDC	0.906	0.960	0.943	0.988	0.810	0.918	0.788	0.916	0.723	0.913	0.725	0.919
DDC-NEG	0.919	0.965	**0.955**	**0.991**	**0.829**	**0.927**	**0.816**	**0.926**	**0.751**	**0.922**	**0.746**	**0.925**

Table 3.4: BCubed F-measure and NMI performance comparisons. For linkage-based approaches, scores are reported using optimal (oracle-supplied) threshold. The best performance is reported in **bold**.

to low precision and high recall operating points. Neither of the two cases provide desirable clustering results. In real-world applications, we are often interested in generating high precision and recall clustering assignments and at the same time know the approximate number of distinct identities. This requires one to find proper operating points. In this section, we investigate the influences of different operating thresholds on the resulting number of clusters. Results on the YTF and LFW datasets are reported. From Figures 3.6a and 3.6b, we observe kinks and clear fall-offs from the proposed methods. The kinks provide hints to the number of distinct identities and reduce the dynamic range of generated number of clusters.

3.6 Conclusion

In this chapter, we proposed a novel algorithm to cluster unconstrained face images without knowing the number of subjects. Based on a local compact representation and a density-based similarity measure, the proposed approach adaptively models the neighborhood structure for each sample and yield a more discriminative



(a) YTF (b) LFW

Figure 3.6: Qualitative evaluations on YTF and LFW.

neighborhood similarity measure. We theoretically showed that the representation is asymptotically a Parzen window density estimator. The proposed approach achieves improved performance than other state-of-the-art approaches on challenging face datasets. The results also show that the density-aware property reduces the difficulty of finding proper operating points in clustering.

Chapter 4: DuDoNet: Dual Domain Network for CT Metal Artifact Reduction

4.1 Overview

Computed tomography (CT) is an imaging modality widely used for medical diagnosis and treatment. CT images are often corrupted by undesirable artifacts when metallic implants are carried by patients, which creates the problem of metal artifact reduction (MAR). Existing methods for reducing the artifacts due to metallic implants are inadequate for two main reasons. First, metal artifacts are structured and non-local so that simple image domain enhancement approaches would not suffice. Second, the MAR approaches that attempt to reduce metal artifacts in the X-ray projection (sinogram) domain inevitably lead to severe secondary artifacts due to sinogram inconsistency. To overcome these difficulties, we propose an end-to-end trainable Dual Domain Network (DuDoNet) to simultaneously restore sinogram consistency and enhance CT images. The linkage between the sinogram and image domains is a novel Radon inversion layer that allows the gradients to back-propagate from the image domain to the sinogram domain during training. Extensive experiments show that our method achieves significant improvements over other single

domain MAR approaches. To the best of our knowledge, it is the first end-to-end dual-domain network for MAR.

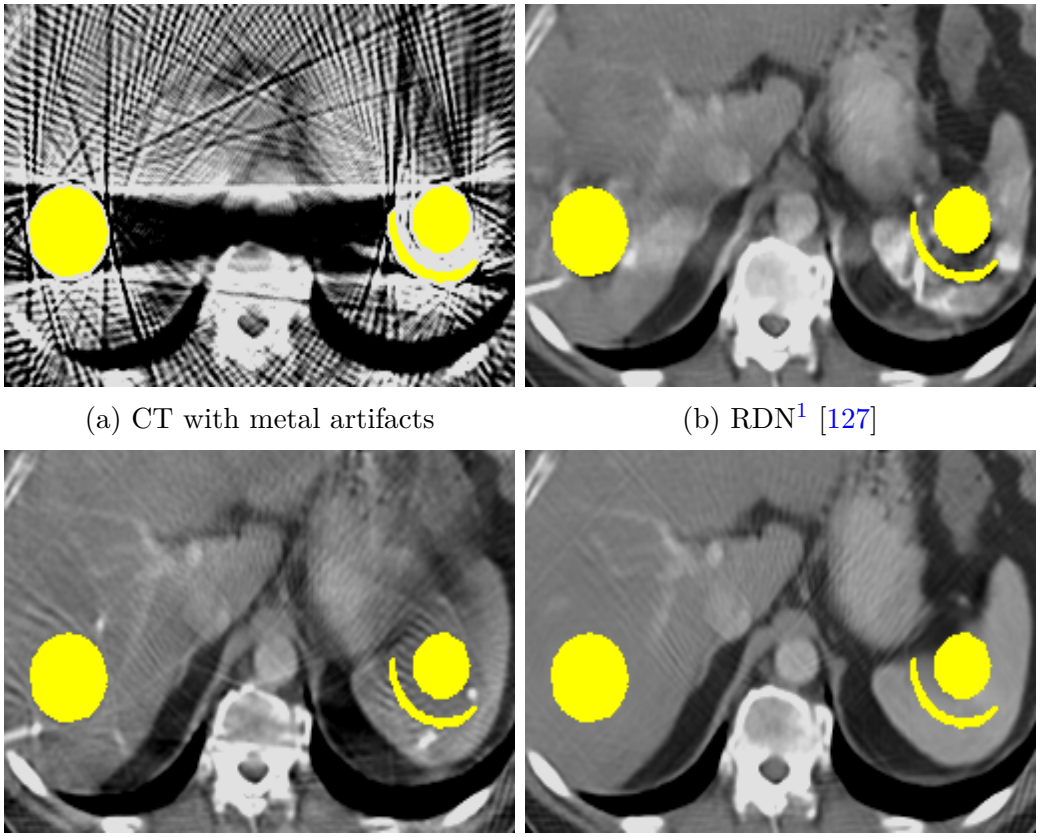
4.2 Introduction

Computed tomography (CT) images reconstructed from X-ray projections allow effective medical diagnosis and treatment. However, due to increasingly common metallic implants, CT images are often adversely affected by metal artifacts which not only exhibit undesirable visual effects but also increase the possibility of false diagnosis. This creates the problem of metal artifact reduction (MAR), for which existing solutions are inadequate.

Unlike typical image restoration tasks such as super-resolution [58, 107, 127, 131], compression artifact removal [33, 125], and denoising [15, 59, 66], metal artifacts are often *structured and non-local* (e.g. streaking and shadowing artifacts as in Figure 4.1a). Modeling such artifacts in image domain is extremely difficult. Therefore, before the emergence of deep learning, most existing works [18, 48, 67, 69] proposed to reduce metal artifact in the X-ray projection (sinogram) domain. The metal-corrupted regions are viewed as missing, and replaced by interpolated values. However, as the projections are taken from a single object under certain geometry, physical constraints should be satisfied by the enhanced sinogram. Otherwise, severe *secondary artifacts* can be introduced in the reconstructed CT images.

Recently, motivated by the success of deep learning in solving ill-posed inverse problems [59, 74, 102, 107, 123, 127], several works have been proposed to overcome

[1]The residual dense network (RDN) proposed in [127] without up-scaling layers.



(a) CT with metal artifacts (b) RDN[1] [127]

(c) CNNMAR [126] (d) DuDoNet (Ours)

Figure 4.1: (a) Sample MAR results for a CT image with intense metal artifacts. Metal implants are colored in yellow. (b) Artifacts are not fully reduced and a 'white band' is present between the two implants. (c) Organ boundaries on the right are smeared out. (d) DuDoNet effectively reduces metal shadows and recovers the fine details.

the difficulties in MAR. Wang et al. [106] applied the pix2pix model [41] to reduce metal artifacts in the CT image domain. Zhang et al. [126] proposed to first estimate a prior image by a convolutional neural network (CNN). Based on the prior image, metal-corrupted regions in the sinogram are filled with surrogate data through several post-processing steps for reduced secondary artifact. Park et al. [76] applied U-Net [83] to directly restore metal-corrupted sinograms. Although metal artifacts can be reduced by these deep learning approaches, we will show that, despite the strong expressive power of deep neural networks, either image domain enhancement

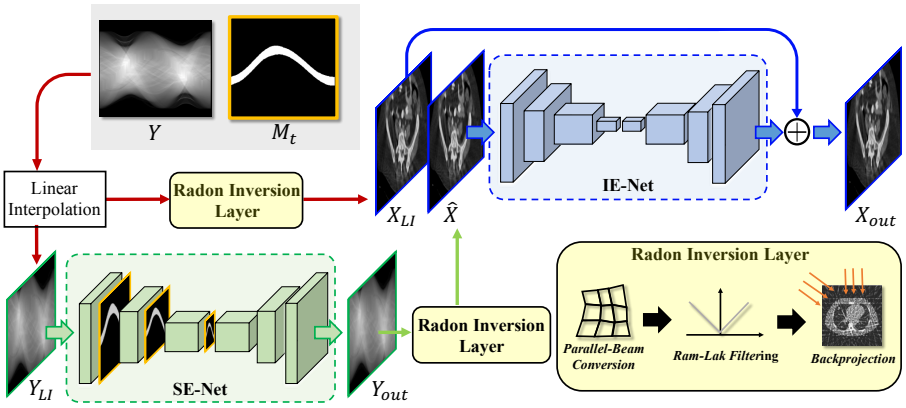


Figure 4.2: The proposed Dual Domain Network (DuDoNet) for MAR. Given a degraded sinogram Y and a metal trace mask $\mathcal{M}_t$, DuDoNet reduces metal artifacts by simultaneously refining in the sinogram and image domains.

or sinogram domain enhancement is limited in being able to restore metal shadows and secondary artifact.

We hereby propose Dual Domain Network (DuDoNet) to address these problems by *learning two CNNs on dual domains to restore sinograms and CT images simultaneously*. Our intuition is that image domain enhancement can be improved by fusing information from the sinogram domain, and inconsistent sinograms can be corrected by the learning signal back-propagated from the image domain to reduce secondary artifacts. Specifically, we propose a novel network (Figure 4.2) consisting of three parts: *a sinogram enhancement network (SE-Net), a Radon inversion layer (RIL), and an image enhancement network (IE-Net)*. To address the issue that in the sinogram domain, information about small metal implants tends to vanish in higher layers of the network due to down-sampling, we propose a mask pyramid U-Net architecture for SE-Net, which retains metal mask information across multiple scales. The key to our dual-domain learning is RIL that reconstructs CT images using the filtered back-projection (FBP) algorithm and efficiently back-propagates

gradients from the image domain to the sinogram domain. Based on RIL, we introduce a Radon consistency loss to penalize secondary artifacts in the image domain. Finally, IE-Net refines CT images via residual learning. Extensive experiments on CT images from hundreds of patients demonstrate that dual domain enhancement generates superior artifact-reduced CT images.

In summary, we make the following contributions:

- We propose an end-to-end trainable dual-domain refinement network for MAR. The network is able to recover details corrupted by metal artifacts.
- We propose a mask pyramid (MP) U-Net to improve sinogram refinement. The MP architecture improves performance especially when small metallic implants are dominated by the non-metal regions.
- We propose a Radon inversion layer (RIL) to enable efficient end-to-end dual domain learning. RIL can benefit the community through its ubiquitous use in various reconstruction algorithms [2, 44, 111, 128].
- We propose a Radon consistency (RC) loss to penalize secondary artifacts in the image domain. Gradients of the loss in the image domain are back-propagated through RIL to the sinogram domain for improved consistency.

4.3 Backgrounds and Related Works

Tissues inside the human body such as bones and muscles have different X-ray attenuation coefficients μ. If we consider a 2D slice of human body, the distribution of the attenuation coefficients $X = \mu(x, y)$ represents the underlying anatomical

structure. The principle of CT imaging is based on the fundamental Fourier Slice Theorem, which guarantees that the 2D function X can be reconstructed solely from its dense 1D projections. In CT imaging, projections of the anatomical structure X are inferred by the emitted and received X-ray intensities through the Lambert-Beer Law [6]. We consider the following CT model under a polychromatic X-ray source with energy distribution $\eta(E)$:

$$Y = -\log \int \eta(E) \exp\{-\mathcal{P}X(E)\}\, dE, \tag{4.1}$$

where $\mathcal{P}$ is the projection generation process, and Y represents the projection data (sinogram). The 2D $X(E)$ is the anatomical structure (CT image) we want to recover from the measured projection data Y .

For normal body tissues, $X(E)$ is almost constant with respect to the X-ray energy E. If we let $X = X(E)$, then

$$Y = \mathcal{P}X. \tag{4.2}$$

Therefore, given measured projection data Y, the CT image $\hat{X}$ can be inferred by using a reconstruction algorithm $\mathcal{P}^{\dagger}$[2]: $\hat{X} = \mathcal{P}^{\dagger}Y$ [47].

However, when metallic implants $I_M(E)$ are present, $X(E) = X + I_M(E)$, where $X(E)$ has large variations with respect to E due to I_M. Eq. (4.1) becomes

$$Y = \mathcal{P}X - \log \int \eta(E) \exp\{-\mathcal{P}I_M(E)\} dE, \tag{4.3}$$

[2]We use $\mathcal{P}^{\dagger}$ to denote the linear operation for reconstruction.

where the region of $\mathcal{P}I_M$ in Y will be referred to as *metal trace* in the rest of the chapter. When the reconstruction algorithm $\mathcal{P}^\dagger$ is applied,

$$\mathcal{P}^\dagger Y = \hat{X} - \mathcal{P}^\dagger \log \int \eta(E) \exp\{-\mathcal{P}I_M(E)\} dE. \tag{4.4}$$

The term after $\hat{X}$ in (4.4) is the metal artifact. It is clear that perfect MAR can be achieved only if the last term in Eq. (4.4) is suppressed while the term $\hat{X}$ is unaffected. However, it is generally an ill-posed problem since both terms contribute to the region of metal trace.

4.3.1 Inpainting-based Methods

One commonly adopted strategy in MAR is to formulate sinogram completion as an image inpainting task. Data within the metal trace are viewed as missing and filled through interpolation. Linear interpolation (LI) [48] is a widely used method in MAR due to its simplicity. Meyer et al. [69] proposed the NMAR algorithm, where sinograms are normalized by tissue priors before performing LI. NMAR requires proper tissue segmentation in the image domain, which is unreliable when severe metal artifacts are present. Mehranian et al. [67] restored sinograms by enforcing sparsity constraints in the wavelet domain. In general, inpainting-based approaches fail to replace the data of $\mathcal{P}X$ in (4.3) within metal trace by consistent values. *It is this introduced inconsistency in sinogram data that leads to noticeable secondary artifacts after reconstruction.*

4.3.2 MAR by Iterative Reconstruction

In iterative reconstruction, MAR can be formulated as the following optimization problem:

$$\hat{X} = \min_{X} \|(1 - \mathcal{M}_t) \odot (\mathcal{P}X - Y)\|^2 + \lambda R(X), \tag{4.5}$$

where $\mathcal{M}_t$ is the metal trace mask. $\mathcal{M}_t = 1$ on the metal trace and $\mathcal{M}_t = 0$ otherwise. R is some regularization function, e.g. total variation (TV) [35] and sparsity constraints in the wavelet domain [122]. Eq. (4.5) is often solved through iterative approaches such as the split Bregman algorithm. Iterative reconstruction usually suffers from long processing time as they require multiplying and inverting huge matrices in each iteration. More importantly, hand-crafted regularization $R(X)$ does not capture the structure of metal artifacts and would result in an over-smoothed reconstruction. Recently, Zhang et al. [122] proposed a re-weighted JSR method which combines NMAR into (4.5) and jointly solves for X and interpolated sinogram. Similar to NMAR, the weighting strategy in re-weighted JSR requires tissue segmentation. In phantom study, better performance against NMAR is achieved by re-weighted JSR. However, the improvements remain limited for non-phantom CT images.

4.3.3 Deep Learning based Methods for MAR

Convolutional neural networks have the ability to model complex structures within data. Motivated by the success of DNNs in solving inverse problems, Gjesteby et al. [28] and Park et al. [76] proposed to refine sinograms using a CNN for improved consistency. Zhang et al. [126] proposed a CNNMAR model to first estimate a prior image by a CNN and then correct sinogram similar to NMAR. However, even with the strong expressive power of CNNs, these approaches still suffer from secondary artifacts due to inconsistent sinograms.

Gjesteby et al. [29], Xu et al. [115] and Wang et al. [106] proposed to reduce metal artifact directly in the CT image domain. The metal artifacts considered in these works are mild and thus can be effectively reduced by a CNN. We will show in our experiments that image domain enhancement is not sufficient for mitigating intense metal shadows.

4.4 Proposed Method

As shown in Figure 2.1, our proposed model consists of three parts: (a) a sinogram enhancement network (SE-Net), (b) a Radon inversion layer (RIL), and (c) an image enhancement network (IE-Net). Inputs to the model include a degraded sinogram $Y \in \mathbb{R}^{H_s \times W_s}$ and the corresponding metal trace mask $\mathcal{M}_t \in \{0,1\}^{H_s \times W_s}$. Notice that we use H_s to represent the detector size and W_s to represent the number of projection views. The region where $\mathcal{M}_t = 1$ is the metal trace. Given the inputs, we first apply LI [48] to generate an initial estimate for the sinogram data within

metal trace. The resulting interpolated sinogram is denoted by Y_{LI}. SE-Net then restores Y_{LI} within the metal trace through a mask pyramid U-Net architecture. To maintain sinogram consistency, we introduce a Radon consistency (RC) loss. A sinogram will be penalized by the RC loss if it leads to secondary artifacts in the image domain after passing through RIL. Finally, the reconstructed CT image $\hat{X} \in \mathbb{R}^{H_c \times W_c}$ is refined by IE-Net via residual learning.

4.4.1 Sinogram Enhancement Network

Sinogram enhancement is extremely challenging since geometric consistency should be retained to prevent secondary artifacts in the reconstructed CT image, so prior works only replace data within the metal trace. Similarly, given a metal-corrupted sinogram Y and metal trace mask $\mathcal{M}_t$, SE-Net $\mathcal{G}_s$ learns to restore the region of Y_{LI} in $\mathcal{M}_t = 1$. In sinogram domain enhancement, when the metal size is small, or equivalently, the metal trace is small, information about metal trace is dominated by non-metal regions in higher layers of network due to down-sampling. To retain the mask information, we propose to fuse $\mathcal{M}_t$ through a mask pyramid U-Net architecture. The output of SE-Net is written as

$$Y_{out} = \mathcal{M}_t \odot \mathcal{G}_s(Y_{LI}, \mathcal{M}_t) + (1 - \mathcal{M}_t) \odot Y_{LI}. \tag{4.6}$$

We use an L_1 loss to train SE-Net:

$$\mathcal{L}_{\mathcal{G}_s} = \|Y_{out} - Y_{gt}\|_1, \tag{4.7}$$

where Y_{gt} is the ground truth sinogram without metal artifact.

4.4.2 Radon Inversion Layer

Although sinogram inconsistency is reduced by SE-Net, there is no existing mechanism to penalize the secondary artifacts in the image domain. The missing key element is an efficient and *differentiable* reconstruction layer. Therefore, we propose a novel RIL f_R to reconstruct CT images from sinograms and at the same time allow back-propagation of gradients. We hightlight that trivially inverting $\mathcal{P}$ in existing deep learning frameworks would require a time and space complexity of $\mathcal{O}(H_s W_s H_c W_c)$, which is prohibitive due to limited GPU memory.

In this work, we consider the projection process $\mathcal{P}$ as the Radon transform under fan-beam geometry with arc detectors [47]. The distance between an X-ray source and its rotation center is D. The resulting fan-beam sinograms Y_{fan} are represented in coordinates (γ, β). To reconstruct CT images from $Y_{fan}(\gamma, \beta)$, we adopt the fan-beam filtered back-projection (FBP) algorithm as the forward operation of RIL.

Our RIL consists of three modules: (a) a parallel-beam conversion module, (b) a filtering module, and (c) a backprojection module. The parallel-beam conversion module transforms $Y_{fan}(\gamma, \beta)$ to its parallel-beam counterpart $Y_{para}(t, \theta)$ through a change of variables. The FBP algorithm in coordinate (t, θ) becomes more effective and memory-efficient than in (γ, β). Parallel-beam FBP is then realized by the subsequent filtering and back-projection modules.

Parallel-beam Conversion Module. We utilize the property that a fan beam sinogram $Y_{fan}(\gamma, \beta)$ can be converted to its parallel beam counterpart $Y_{para}(t, \theta)$ through the following change of variables [47]:

$$\begin{cases} t = D \sin \gamma, \\ \theta = \beta + \gamma. \end{cases} \tag{4.8}$$

The change of variable in (4.8) is implemented by grid sampling in (t, θ), which allows back-propagation of gradients. With Y_{para}, CT images can be reconstructed through the following Ram-Lak filtering and back-projection modules.

Ram-Lak Filtering Module. We apply the Ram-Lak filtering step to Y_{para} in the Fourier domain.

$$Q(t, \theta) = \mathcal{F}_t^{-1} \left\{ |\omega| \cdot \mathcal{F}_t \left\{ Y_{para}(t, \theta) \right\} \right\}, \tag{4.9}$$

where $\mathcal{F}_t$ and $\mathcal{F}_t^{-1}$ are the Discrete Fourier Transform (DFT) and inverse Discrete Fourier Transform (iDFT) with respect to the detector dimension.

Backprojection Module. The filtered parallel-beam sinogram Q is back-projected to the image domain for every projection angle θ by the following formula:

$$X(u, v) = \int_0^{\pi} Q(u \cos \theta + v \sin \theta, \theta) d\theta. \tag{4.10}$$

It is clear from (4.10) that the computation is highly parallel. We make a remark here regarding the property of RIL f_R. Due to the back-projection nature of f_R,

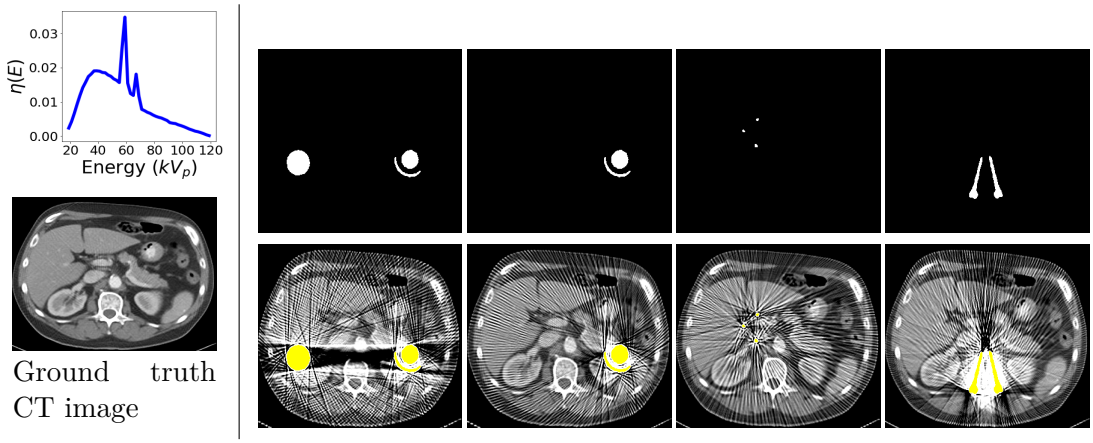


Figure 4.3: Sample simulated metal artifact on patient CT. The X-ray spectrum is shown in the upper-left corner. Metallic implants are colored in yellow for better visualization.

the derivative with respect to the input Y_{out} is actually the projection operation $\mathcal{P}$. That is, any loss in the image domain will be aggregated and projected to the sinogram domain. This desirable property enables joint learning in sinogram and image domains.

Radon Consistency Loss. With the differentiable RIL, we introduce the following Radon consistency (RC) loss to penalize secondary artifacts in $\hat{X} = f_R(Y_{out})$ after reconstruction.

$$\mathcal{L}_{RC} = \|f_R(Y_{out}) - X_{gt}\|_1, \tag{4.11}$$

where X_{gt} is the ground truth CT image without metal artifact.

Difference from DL-based Reconstruction. Our RIL is designed to combine the image formation process (CT reconstruction) with deep neural networks and achieve improved MAR by dual-domain consistency learning. Methods in [2, 44, 111, 128] target *image formation via deep learning*, which is not the main

focus of this work.

4.4.3 Image Enhancement Network

Since our ultimate goal is to reduce visually undesirable artifacts in image domain, we further apply a U-Net $\mathcal{G}_i$ to enhance $\hat{X}$ by residual learning:

$$X_{out} = X_{LI} + \mathcal{G}_i(\hat{X}, X_{LI}), \tag{4.12}$$

where $X_{LI} = f_R(Y_{LI})$ is reconstructed from Y_{LI}, the linearly interpolated sinogram. $\mathcal{G}_i$ is also optimized by L_1 loss.

$$\mathcal{L}_{\mathcal{G}_i} = \|X_{out} - X_{gt}\|_1. \tag{4.13}$$

The full objective function of our model is:

$$\mathcal{L} = \mathcal{L}_{\mathcal{G}_s} + \mathcal{L}_{RC} + \mathcal{L}_{\mathcal{G}_i}. \tag{4.14}$$

One could tune and balance each term in (4.14) for better performance. However, we found that the default setting works sufficiently well.

4.5 Experimental Results

Following the de facto practice in the literature [126], our evaluations consider simulated metal artifacts on real patient CTs. Various effects are considered includ-

ing polychromatic X-ray, partial volume effect, and Poisson noise. The simulated artifacts exhibit complicated structures and cannot be easily modelled by a very deep CNN. All the compared approaches are evaluated on the same dataset, and superior performance is achieved by our method.

Metal Artifact Dataset. Recently, Yan et al. [116] released a large-scale CT dataset DeepLesion for lesion detection. Due to its high diversity and quality, we use a subset of images from DeepLesion to synthesize metal artifact. 4,000 images from 320 patients are used in the training set and 200 images from 12 patients are used in the test set. All images are resized to 416×416. We collect a total of 100 metal shapes. 90 metal shapes are paired with the 4,000 images, yielding 360,000 combinations in the training set. 10 metal shapes are paired with the 200 images, yielding 2,000 combinations in the test set. In the training set, the sizes of the metal implants range from 16 to 4967 pixels. In the test set, the sizes of the metal implants range from 32 to 2054 pixels.

We adopt similar procedures as in [126] to synthesize metal-corrupted sinograms and CT images. We assume a polychromatic X-ray source with spectrum $\eta(E)$ in Figure 4.3. To simulate Poisson noise in the sinogram, we assume the incident X-ray has 2×10^7 photons. Metal partial volume effect is also considered. The distance from the X-ray source to the rotation center is set to 39.7cm, and 320 projection views are uniformly spaced between 0-360 degrees. The resulting sinograms have size 321×320. Figure 4.3 shows some sample images with simulated metal artifacts.

Evaluation Metrics. We choose peak signal-to-noise ratio (PSNR) and

	Large Metal ⟶ Small Metal					Average
A)	22.88/0.7850	24.52/0.8159	27.38/0.8438	28.61/0.8549	28.93/0.8581	26.46/0.8315
B)	23.06/0.7868	24.71/0.8178	27.66/0.8463	28.91/0.8575	29.19/0.8604	26.71/0.8337
C)	27.54/0.8840	29.49/0.9153	31.96/0.9368	34.38/0.9498	33.90/0.9489	31.45/0.9269
D)	28.46/0.8938	30.67/0.9232	33.71/0.9458	36.17/0.9576	35.74/0.9571	32.95/0.9355
E)	28.28/0.8921	30.49/0.9221	33.76/0.9456	36.26/0.9576	36.01/0.9574	32.96/0.9350
F)	28.97/0.8970	31.14/0.9254	34.21/0.9476	36.58/0.9590	36.15/0.9586	33.41/0.9375
G)	29.02/0.8972	31.12/0.9256	34.32/0.9481	36.72/0.9595	36.36/0.9592	33.51/0.9379

Table 4.1: Quantitative evaluations for different components in DuDoNet. (PSNR/SSIM)

structured similarity index (SSIM) for quantitative evaluations. In DeepLesion, each CT image is provided with a dynamic range, within which the tissues are clearly discernible. We use the dynamic range as the peak signal strength when calculating PSNR.

Implementation Details. We implement our model using the PyTorch [77] framework. All the sinograms have size 321×320, and all the CT images have size 416×416. To train the model, we use the Adam [51] optimizer with $(\beta_1, \beta_2) = (0.5, 0.999)$, and a batch size of 8. The learning rate starts from 2×10^{-4}, and is halved for every 30 epochs. The model is trained on two Nvidia 1080Ti for 380 epochs.

4.5.1 Ablation Study

In this section, we evaluate the effectiveness of different components in the proposed approach. Performance is evaluated on the artifact-reduced CT images. When evaluating SE-Nets without image domain refinement, we use the reconstructed CT images $\hat{X}$. We experiment on the following configurations:

A) SE-Net$_0$: The sinogram enhancement network without mask pyramid network.

B) SE-Net: The full sinogram enhancement module.

C) IE-Net: Image enhancement module. IE-Net is applied to enhance X_{LI} without $\hat{X}$.

D) SE-Net$_0$+IE-Net: Dual domain learning with SE-Net$_0$ and IE-Net.

E) SE-Net+IE-Net: Dual domain learning with SE-Net and IE-Net.

F) SE-Net$_0$+IE-Net+RCL: Dual domain learning with Radon consistency loss.

G) SE-Net+IE-Net+RCL: Our full network.

Notice that the configurations including SE-Net$_0$, SE-Net and IE-Net are single domain enhancement approaches.

Table 4.1 summarizes the performance of different models. Since there are totally 10 metal implants in the test set, for conciseness, we group the results according to the size of metal implants. The sizes of the 10 metal implants are: [2054, 879, 878, 448, 242, 115, 115, 111, 53, 32] in pixels. We simply put every two masks into one group.

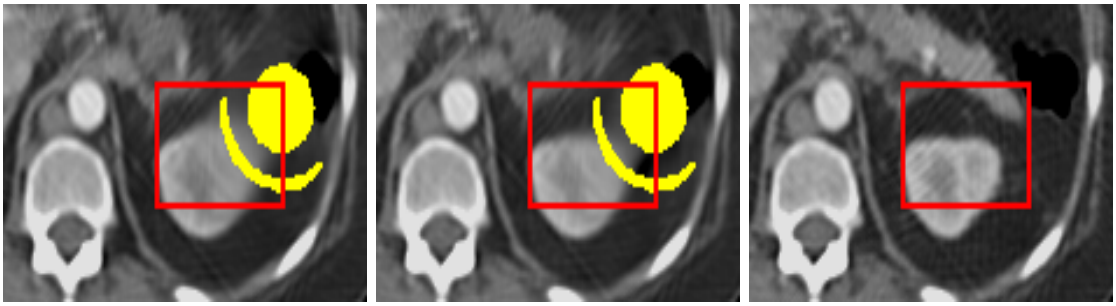
From E and G, it is clear that the use of the RC loss improves the performance over all metal sizes for at least 0.3 dB. In Figure 4.4, the model trained with RC loss better recovers the shape of the organ.

From F and G, we observe an interesting trend that the proposed mask pyramid architecture results in $\sim$0.2 dB gain when the metal size is small, and the performance is nearly identical when the metal is large. The reason is that the

PSNR/SSIM	Large Metal ⟶ Small Metal				
LI [48]	20.20/0.8236	22.35/0.8686	26.76/0.9098	28.50/0.9252	29.53/0.9312
NMAR [69]	21.95/0.8333	24.43/0.8813	28.63/0.9174	30.84/0.9281	31.69/0.9402
cGAN-CT [106]	26.71/0.8265	24.71/0.8507	29.80/0.8911	31.47/0.9104	27.65/0.8876
RDN-CT [127]	28.61/0.8668	28.78/0.9027	32.40/0.9264	34.95/0.9446	34.00/0.9376
CNNMAR [126]	23.82/0.8690	26.78/0.9097	30.92/0.9394	32.97/0.9513	33.11/0.9520
DuDoNet (Ours)	**29.02/0.8972**	**31.12/0.9256**	**34.32/0.9481**	**36.72/0.9595**	**36.36/0.9592**

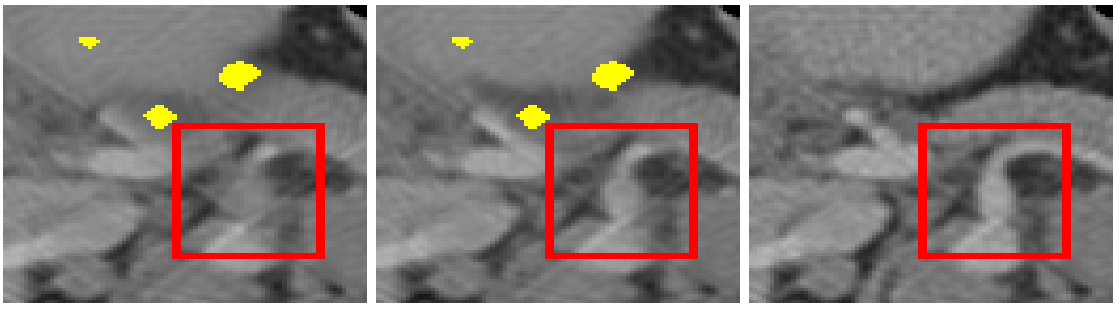
Table 4.2: Quantitative evaluation of MAR approaches in terms of PSNR and SSIM.

mask pyramid retains metal information across multiple scales. Figure 4.5 demonstrates that in the proximity of small metal implants, the model with mask pyramid recovers the fine details.



Without RC loss With RC loss Ground Truth

Figure 4.4: Visual comparisons between models without RC loss (E in Table 4.1) and our full model (G in Table 4.1).



Without MP With MP Ground Truth

Figure 4.5: Visual comparisons between models without MP (F in Table 4.1) and our full model (G in Table 4.1).

Effect of Dual Domain Learning. In the proposed framework, IE-Net enhances X_{LI} by fusing information from SE-Net. We study the effect of dual

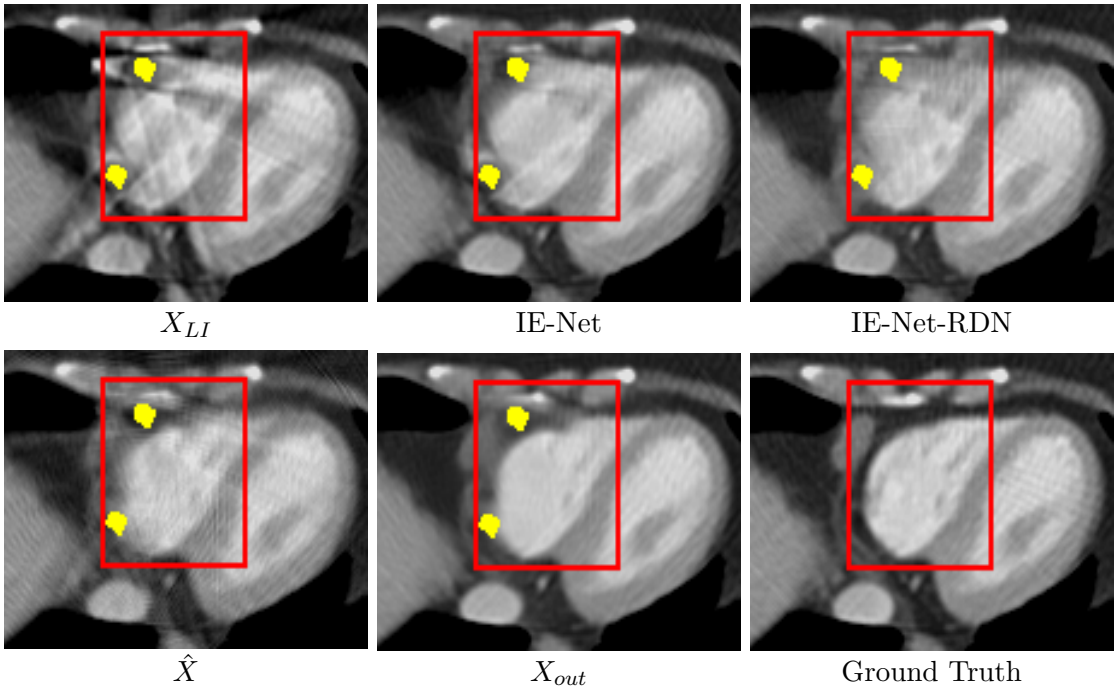
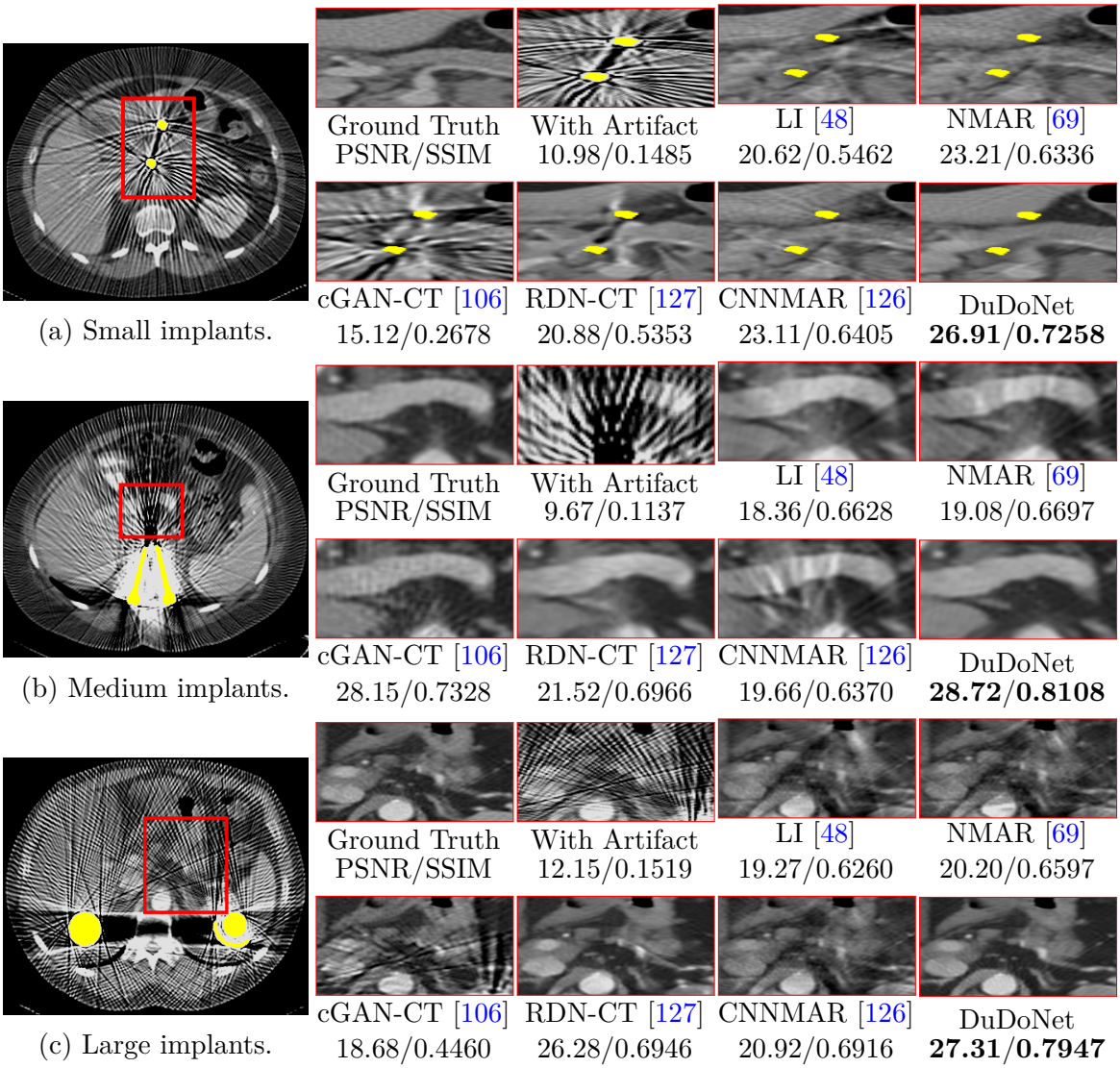


Figure 4.6: Visual comparisons between models without SE-Net (top row IE-Net and IE-Net-RDN) and our full model (bottom row $\hat{X}$ and X_{out}).

domain learning by visually comparing our full pipeline (G in Table 4.1) with single domain enhancement IE-Net (C in Table 4.1). In addition to the U-Net architecture, we also consider IE-Net with RDN architecture, which is denoted as IE-Net-RDN. Visual comparisons are shown in Figure 4.6. We observe that single domain models IE-Net and IE-Net-RDN fail to recover corrupted organ boundaries in X_{LI}. In our dual domain refinement network, SE-Net first recovers inconsistent sinograms and reduces secondary artifacts as in $\hat{X}$. IE-Net then refines $\hat{X}$ to recover the fine details.

Effect of LI sinogram. The inputs to our network are the linear interpolated sinogram Y_{LI} and its reconstructed CT X_{LI}. One possible alternative is to directly input the metal corrupted sinogram and CT, and let the network learn to restore the intense artifacts. However, we experimentally found out this alternative approach does not perform well. Metal shadows and streaking artifacts are not fully



(a) Small implants.

(b) Medium implants.

(c) Large implants.

Figure 4.7: Visual comparisons on MAR for different types of metallic implants.

suppressed.

4.5.2 Comparison with State-of-the-Art Methods

In this section, we compare our model with the following methods: LI [48], NMAR [69], cGAN-CT [106], RDN-CT [127] and CNNMAR [126]. We use cGAN-CT to refer the approach by Wang et al. [106] which applies cGAN for image domain MAR. RDN [127] was originally proposed for image super-resolution (SR). The fun-

damental building unit of RDN is the residual dense block (RDB). Recently, it has been shown that by stacking multiple RDBs or its variant, the residual in residual dense blocks (RRDBs) [107], local details in natural images can be effectively recovered. We build a very deep architecture with 10 RDBs (∼80 conv layers) for direct image domain enhancement, which is denoted by RDN-CT. Specifically, we select $D = 10, C = 8, G = 64$, following the notations in [127]. Inputs to RDN-CT are 128×128 patches.

Quantitative Comparisons. Table 4.2 shows quantitative comparisons. We observe that the state-of-the-art sinogram inpainting approach CNNMAR achieves higher SSIM than image enhancement approaches (e.g. RDN and cGAN-CT) especially when the size of metal is small. The reason is that sinogram inpainting only modifies data within the metal trace and recovers the statistics reasonably well. In most of the cases, CNNMAR also outperforms cGAN-CT in terms of PSNR. However, when CNN is sufficiently deep (e.g. RDN-CT), image enhancement approaches generally achieve higher PSNR. Our dual domain learning approach jointly restores sinograms and CT images, which attains the best performance in terms of both PSNR and SSIM *consistently in all categories*.

Visual Comparisons. Figure 4.7 shows visual comparisons. Figure 4.7a considers metal artifacts resulted from two small metallic implants. From the zoomed figure (with metal artifact), we can perceive severe streaking artifacts and intense metal shadows between the two implants. We observe that sinogram inpainting approaches such as LI, NMAR and CNNMAR effectively reduce metal shadows. However, fine details are either corrupted by secondary artifacts as in LI or blurred

as in NMAR and CNNMAR. Image domain approaches such as cGAN-CT and RDN-CT produce sharper CT images but fail to suppress metal shadows. Our method effectively reduces metal shadows and at the same time retains fine details. Figure 4.7b shows a degraded CT image with long metal implants. We observe similar trend that sinogram inpainting approaches do not perform well in regions with intense streaking artifact. In this example, image domain methods reduce most of the artifacts. It is possibly due to that fact that the pattern of the artifact in Figure 4.7b is monotonous compared to Figures 4.7a and 4.7c. However, noticeable speckle noise is present in the result by cGAN-CT, and RDN-CT does not fully recover details in the middle. Figure 4.7c considers metal artifacts result from two large metallic implants. Likewise, sinogram inpainting methods and direct image domain enhancement have limited capability of suppressing metal artifacts.

Evaluations on CT Images with Real Metal Artifacts. Evaluating MAR methods on CT images of patients carrying metal implants is challenging for two reasons: (1) Modern clinical CT machines have certain build-in MAR algorithms. Evaluations on CT images after MAR would not be meaningful; (2) Sinogram data with metal artifacts are difficult to access, except perhaps from machine manufacturers. To the best of our knowledge, there is no existing sinogram database which targets MAR.

In order to compare different MAR methods, we manually collect CT images with metal artifact from DeepLesion [116] and apply the following steps to obtain the metal trace $\mathcal{M}_t$ and the LI sinogram Y_{LI}. DuDoNet can be applied by taking $\mathcal{M}_t$ and Y_{LI} as inputs. Conceptually, the steps can be understood as projecting

the input CT image with unknown imaging geometry to the source domain[3] with known geometry.

(i) $\mathcal{M}_t$: We first segment out the metal mask by applying a threshold of 2,000 HU to the metal-corrupted CT image. $\mathcal{M}_t$ can be obtained by forward projection with the imaging geometry presented in Section 4 in the manuscript.

(ii) Y_{LI}: We adopt the same simulation procedures and imaging geometry as in the manuscript to synthesize metal-corrupted sinogram Y. Y_{LI} can be generated from Y and $\mathcal{M}_t$ by linear interpolation.

Figure 4.8 presents visual comparisons of different MAR algorithms. Metal masks obtained by step (i) are colored in yellow. We would like to emphasize that the true sinogram of a given CT image cannot be inferred without information about the actual imaging geometry (e.g. source to detector distance, and number of projection views). Therefore, in Figure 4.8, due to inconsistent imaging geometry, sinogram-based MAR approaches (e.g. LI) may lead to an even worse visual quality than raw CT. In contrast, DuDoNet effectively reduces metal artifacts.

4.5.3 Running Time Comparisons

On an Nvidia 1080Ti GPU, it takes 0.24 ms for RIL to reconstruct a sinogram of size 321×320 to a CT image of size 416×416, and 11.40 ms for back-propagation of gradients. RIL requires 16 MB of memory for forward pass and 25 MB for back-propagation. In Table 4.3 we compare the running time of different MAR approaches. With the running time of LI included, DuDoNet runs almost $4\times$ faster

[3]The domain of CT images with simulated metal artifacts.

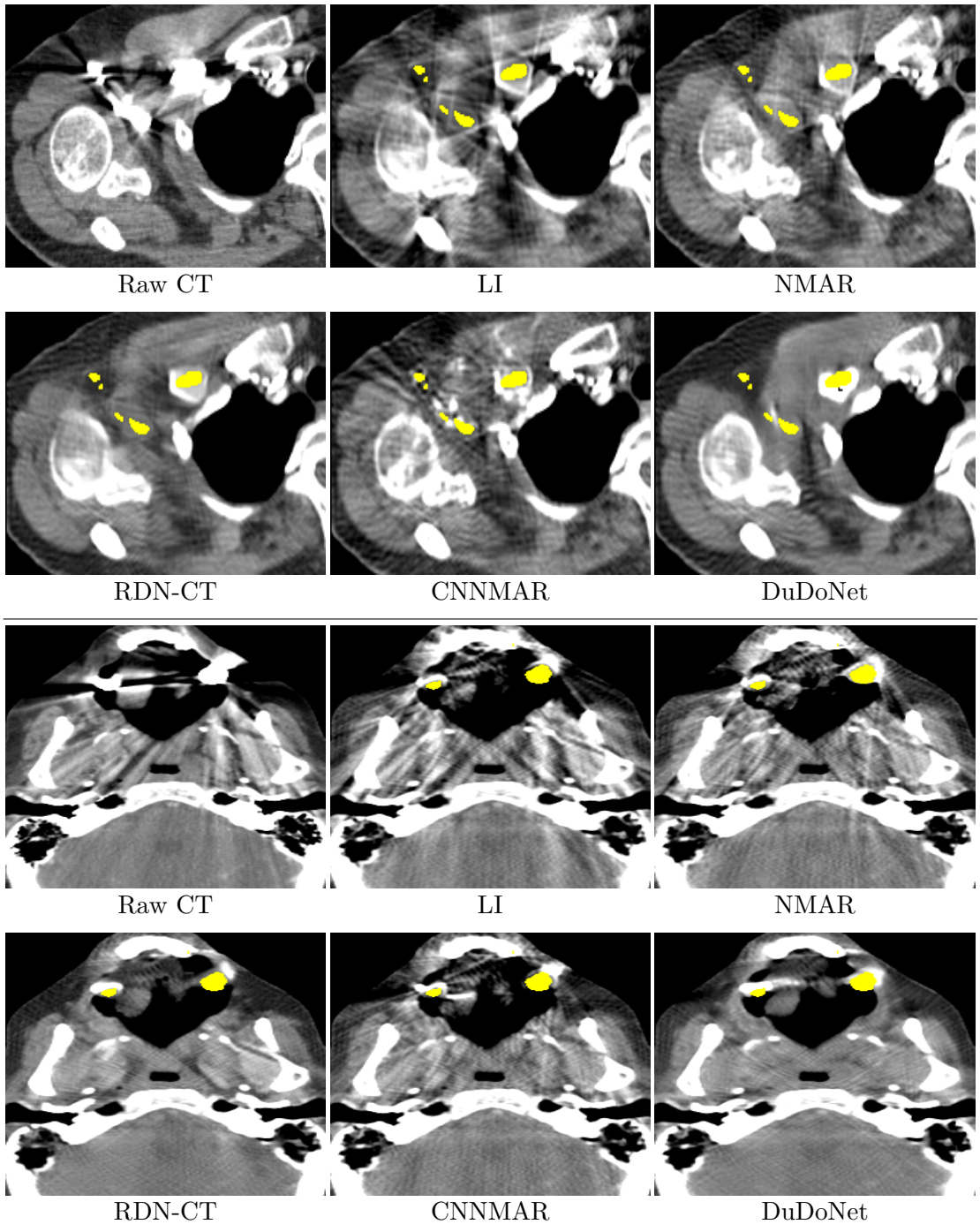


Figure 4.8: Evaluations on CT images with real metal artifacts.

than the very deep architecture RDN while achieving superior performance.

LI [48]	NMAR [69]	cGAN-CT [106]	RDN-CT [127]	CNNMAR [126]	DuDoNet (Ours)
0.0832	0.4180	0.0365	0.5150	0.6043	0.1335

Table 4.3: Comparison of running time measured in seconds.

4.6 Conclusion

In this chapter, we presented the Dual Domain Network for metal artifact reduction. In particular, we proposed to jointly improve sinogram consistency and refine CT images through a novel Radon inversion layer and a Radon consistency loss, along with a mask pyramid U-Net. Experimental evaluations demonstrate that while state-of-the-art MAR methods suffer from secondary artifacts and very-deep neural networks have limited capability of directly reducing metal artifacts in image domain, our dual-domain model can effectively suppress metal shadows and recover details for CT images. At the same time, our network is computationally more efficient. Future work includes investigating the potential of the dual-domain learning framework for other signal recovery tasks, such as super-resolution, noise reduction, and CT reconstruction from sparse X-ray projections.

Chapter 5: Invert and Defend: Model-based Approximate Inversion of Generative Adversarial Networks for Secure Inference

5.1 Overview

Inferring the latent variable that generates a given test sample is a challenging problem in Generative Adversarial Networks (GANs). In this chapter, we propose InvGAN - a novel framework for solving the inference problem in GANs, which involves training an encoder network capable of inverting a pre-trained generator network without access to any training data. Under mild assumptions, we theoretically show that using InvGAN, we can approximately invert the generations of any latent code of a trained GAN model. Furthermore, we empirically demonstrate the superiority of our inference scheme by quantitative and qualitative comparisons with other methods that perform a similar task. We also show the effectiveness of our framework in the problem of adversarial defenses where InvGAN can successfully be used as a projection-based defense mechanism. Experimental validation on several benchmark datasets demonstrate the efficacy of our method in achieving improved performance on several white-box attacks.

5.2 Introduction

Generative Adversarial Networks (GANs) have shown to be successful for generative modeling. Significant research progress in GANs over the last few years has pushed boundaries in generation capabilities and has made possible the synthesis of photo-realistic images of human faces [49, 50] and objects [9]. A fundamental problem involving GANs is the problem of inversion – given a test image, what is the most likely latent code that generates the test sample? The inversion problem is extremely challenging since the generator network in GANs is highly non-linear and non-injective. Inversion has applications in several machine learning problems e.g. domain adaptation [1, 8], compressed sensing [7], adversarial defenses [85], and anomaly detection [87].

In this chapter, we propose a novel approach for addressing the inversion problem in GANs. Our approach is model-based where the mapping from image space to latent space is represented as a parametric function. We solve for the parameters of this function by sampling the latent codes from the noise distribution of the GAN and making sure that (a) the inversions produced from the generated samples are close to the sampled codes (b) the generated images of the inversions semantically match the GAN generations and (c) the distribution of inverted images follows the distribution of the GAN. Our method is a data-free inversion mechanism i.e., given a pre-trained generator network, no access to input dataset is needed. This is particularly important in privacy-preserving learning scenarios in which the data provider does not intend to publicly release the data due to privacy reasons,

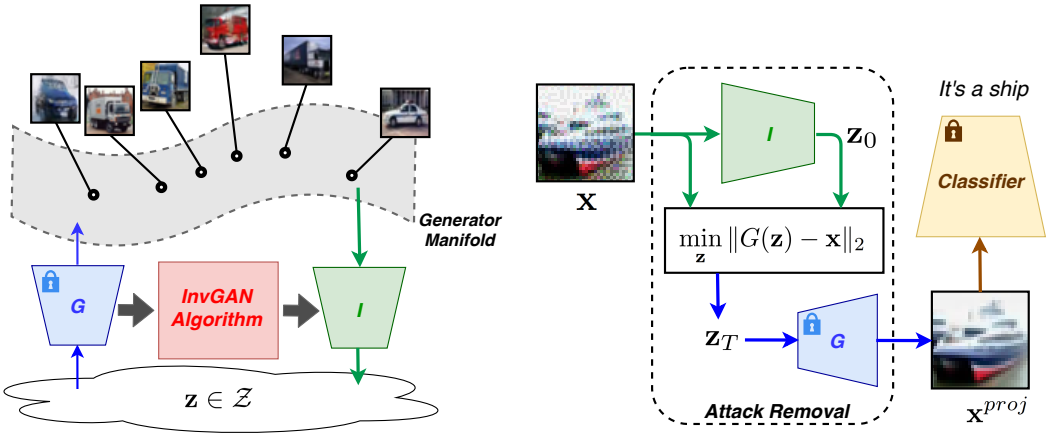


Figure 5.1: Overview of the proposed InvGAN framework. Left: Given a pre-trained generator G and no data, we solve for I to approximately invert the generator. Right: The application of InvGAN in adversarial defenses, where InvGAN can be used to project an adversarially perturbed sample $\mathbf{x}$ onto the generator manifold, and the projected sample $\mathbf{x}^{proj}$ can be used to make a robust prediction.

but instead releases a GAN model trained on this data satisfying several privacy constraints [114]. Our approach can invert such a GAN model.

In addition to comparing the reconstruction performance with previously proposed encoder-GAN models, we demonstrate the effectiveness of our inversion approach for the problem of adversarial defenses. The vulnerability of deep neural networks to small imperceptible perturbations has been demonstrated in several recent papers [11, 30, 71, 97], and this poses a huge threat in security-critical application domains where these networks are used.

One of the recent defense strategies proposed is DefenseGAN [85], which uses a GAN as a defense mechanism. In DefenseGAN, the inference step is used to project a test sample onto the GAN's manifold to remove possible attacks. The inversion is posed as a non-convex optimization problem which is solved using gradient descent. However, this needs careful hyper-parameter tuning per dataset and

generator, is extremely slow in practice, and does not scale well for deeper generator models. In this chapter, we propose an inference procedure to speed up DefenseGAN. More specifically, we use our trained encoder network to initialize the optimization problem. Using this initialization, the inference problem can be solved in very few iterations while preserving the quality of reconstructions. This leads to effortless hyperparameter tuning, a dramatic speed up in runtime, and improved reconstruction results.

5.3 Backgrounds and Related Works

5.3.1 Inverting Generative Models

While significant research has focused on improving the quality, stability and diversity of GANs [9, 49, 50], there has been relatively less work on the inversion problem despite its practical significance. The method in [61] poses inversion as a non-convex optimization problem, which is then solved using projected gradient descent with stochastic clipping. A similar optimization with logistic loss has been proposed in [14]. While the above two methods are model-free and work for a pre-trained generator, they are extremely slow and deliver poor reconstructions for harder datasets like CIFAR-10.

Another line of work involves modifying the GAN objective to support generation and inference in a unified framework. They typically involve training an encoder function that maps from the image space to the latent space jointly with the generator function in GANs. Methods presented in [16, 19] trains the encoder

network using an adversarial loss on (latent, image) pairs. The method proposed in [103] uses adversarial and reconstruction loss in latent space to train the encoder. While these methods enable fast inference, modifying the GAN objective to support inference affects the quality of generator models. Our approach offers the best of both worlds – fast inference and ability to perform inference on a pre-trained generator, thus preserving the quality of generative models.

5.3.2 Adversarial Attacks and Defenses

Adversarial attacks are imperceptible changes crafted to the input samples by an adversary to flip a model's prediction $\hat{y} = f(x)$. In this work, we focus on classification problems, hence, $f(x) \in \{1, \cdots, c\}$. The symbol J represents the classification loss function, Z is the output of the logit layer, and y is the original label. The most common form of adversarial attacks are additive perturbations where a norm-bounded perturbation δ is added to the input sample $x \in \mathbb{R}^{w \times h \times c}$ as $x^{adv} = x + \delta$. A wide range of adversarial attack methods have been proposed.

Fast Gradient Sign Method (FGSM). FGSM is the most simple form of adversarial attack which maximizes the loss function along the gradient direction:

$$x^{adv} = x + \epsilon \cdot sign(\nabla J_y(x)). \tag{5.1}$$

Projected Gradient Descent (PGD). PGD is a simple variant of FGSM by

applying it multiple times with a smaller step size α.

$$x_0^{adv} = x, \quad x_{t+1}^{adv} = x_t^{adv} + \text{clip}_\epsilon \left[\alpha \cdot sign(\nabla J_y(x_t^{adv})) \right]. \tag{5.2}$$

Carlini and Wagner's L2 Attack (CW).

$$\|x^{adv} - x\| + c \cdot l_{CW}(x^{adv}, y), \tag{5.3}$$

where $l_{CW}(x^{adv}, y) = \max(\max\{Z(x^{adv})_i : i \neq y\} - Z(x^{adv})_y, -\tau)$.

To protect the classifiers from adversarial attacks, one line of research focuses on removing the perturbation from input samples before feeding them to the classifier. MagNet [68] detects and reforms adversarial images using autoencoders. DefenseGAN [85] uses a generative adversarial network to model the image manifold. Adversarial perturbations are removed by projecting the samples onto the learned manifold. A similar idea has been used in PixelDefend [96] where the input distribution is modeled using a PixelCNN, and adversarial perturbations are removed from the input samples using greedy decoding.

5.3.3 Circumventing Obfuscated Gradients

These projection-based defense methods take advantage of the operations that leads to *obfuscated gradients* which results in the inability to derive gradients to craft white-box attacks. Several recent works have been proposed to craft adversarial images even when direct gradient calculation is not feasible.

Backward Pass Differentiable Approximation (BPDA) [4]. BPDA approximates the non-differentiable operations with surrogates (e.g., identity function) during backpropagation. Adversarial attacks can then be crafted on the target classifiers using the gradients derived from the surrogates.

Overpowered Attack [42]. Overpowered Attack assumes that a generative model $G(z)$ which approximates the true data distribution is available. On-manifold adversarial attacks can be found by solving the following optimization problem:

$$z^* = \underset{z \in \mathbb{R}^d}{\operatorname{argmax}} \min_{\lambda \geq 0} \mathbb{E}_{\tau \sim \mathcal{N}(0,I)} \left[l_{CW} \left(G(z) + \sigma \frac{\tau}{\|\tau\|_2}, y \right) \right] + \lambda \left(\epsilon - \frac{\|G(z) - x\|_2}{h \cdot w} \right), \tag{5.4}$$

where ϵ is the perturbation budget of the attack. The adversarial example is given by $x^{adv} = G(z^*)$. Unlike common norm-bounded attacks where the perturbation $x^{adv} - x$ is a simple transformation of the loss gradients, adversarial examples crafted by Overpowered Attack can have different semantics from the original image if the norm budget ϵ is not sufficiently small.

5.4 Proposed Method

The generator network in a GAN model $G(z) : \mathbb{R}^d \to \mathbb{R}^{w \times h \times c}$ maps a latent code $z \in \mathbb{R}^d$ to an image in $\mathbb{R}^{w \times h \times c}$. The inversion problem is to find the inverse mapping i.e., given a test sample x, we are interested in finding a latent $\hat{z}$ such that $G(\hat{z}) \approx x$. This problem is extremely hard as most generator networks used in practice are non-convex and non-bijective functions. One common approach to this

problem [61, 85] involves solving the following optimization problem:

$$\min_{z} \|G(z) - x\|_2 \tag{5.5}$$

This optimization is extremely hard due to the non-convexity of the objective function. Solving this requires multiple random initialization of z, carefully tuned learning rate and number of optimization steps for each dataset. Also, this optimization is extremely slow, and scales poorly with increasing complexity of the generator network and the input distribution.

To fix these issues, we propose using an inverter network $I_{\theta_I}(x) : \mathbb{R}^{w \times h \times c} \to \mathbb{R}^d$ that maps a sample from the image space to the latent space as an initialization for z in (5.5). The goal of the inverter network is to approximately invert the generator network. We would like to emphasize that exact inversion is not possible as the generator function is non-bijective.

Theorem 5.1 *Let $\{G(z^{(i)})\}_{i=1}^N$ represent the generated samples corresponding to a pre-trained generator function G, with the noise vectors $z^{(i)} \sim \mathcal{N}(0, 1)$. Let $I(\cdot)$ represent an inverter function that is trained to achieve approximate inversion on the training set, i.e.,*

$$\|I(G(z^{(i)})) - z^{(i)}\|_2 < \epsilon, \quad \forall z^{(i)}, i \in [n].$$

Let L be the Lipschitz constant corresponding to the composite function $I \circ G(\cdot)$.

Then, for $\epsilon' > \epsilon$, with probability $1 - o(1)$,

$$\|I(G(z)) - z\|_2 < \epsilon', \quad for \;\; z \sim \mathcal{N}(0, I).$$

That is with high probability, the function $I(\cdot)$ approximately inverts the generator $G(\cdot)$.

Proof.

The input samples $\{G(z^{(i)})\}_{i=1}^{N}$ correspond to the training data for the inverter network. For any latent $z \sim \mathcal{N}(0, I)$,

$$P(\|z - z^{(i)}\| < \epsilon) \geq 1 - e^{-\frac{d}{18}(\frac{\epsilon^2}{4d} - 1)^2} \quad \forall i \in [n].$$

The above inequality follows from the concentration bound for χ^2 distribution [105] since $\frac{1}{4}\|z - z^{(i)}\|^2$ follows a χ^2_d distribution with d degrees of freedom, where d is the noise dimension. Then,

$$P(\exists z^{(i)}, i \in [n] \text{ s.t } \|z - z^{(i)}\| < \epsilon) \geq 1 - e^{-\frac{nd}{18}(\frac{\epsilon^2}{4d} - 1)^2}. \tag{5.6}$$

Eq. (5.6) says that there exists at least one $z^{(i)}$ concentrated close to z. Now,

consider $\|I(G(z)) - z\|$. This can be expanded as

$$\begin{aligned}
\|I(G(z)) - z\| &= \|(I(G(z)) - I(G(z^{(i)}))) + (I(G(z^{(i)})) - z^{(i)}) + (z^{(i)} - z)\| \\
&\leq \|I(G(z)) - I(G(z^{(i)}))\| + \|I(G(z^{(i)})) - z^{(i)}\| + \|z^{(i)} - z\| \\
&\leq (L+1)\|z - z^{(i)}\| + \|I(G(z^{(i)})) - z^{(i)}\| \\
&\leq (L+1)\|z - z^{(i)}\| + \epsilon, \quad \forall i \in [n] \\
&\leq \min_{i \in [n]} (L+1)\|z - z^{(i)}\| + \epsilon. \qquad (5.7)
\end{aligned}$$

This follows from the triangle inequality and the assumption on the training loss. Using (5.6) and (5.7) for bounding $\|I(G(z)) - z\|$, we obtain

$$\begin{aligned}
P(\|I(G(z)) - z\| \leq \epsilon') &\geq P\Big([\min_{i \in [n]} (L+1)\|z - z^{(i)}\|] < \epsilon' - \epsilon\Big) \\
&= P(\exists i \ \|z - z^{(i)}\| < \frac{\epsilon' - \epsilon}{L+1}) \\
&\geq 1 - e^{-\frac{nd}{18}(\frac{(\epsilon'-\epsilon)^2}{4d(L+1)^2} - 1)^2}.
\end{aligned}$$

That is with probability $1 - o(1)$, $\|I(G(z)) - z\| \leq \epsilon'$. Please note that we assumed that $\epsilon' > \epsilon$. This concludes the proof.

The above theorem states that under some smoothness conditions on the generator-encoder pair, if the encoder loss is bounded for every training sample, then the encoder approximately inverts the generator with high probability.

5.4.1 Encoder Training

A natural way to train the encoder is to minimize the following loss function:

$$\min_{\theta_I} \mathbb{E}_{x \in p_{data}} \|x - G(I_{\theta_I}(x))\|_2^2. \tag{5.8}$$

One issue with this objective arises from the non-surjective nature of the the generator network. There are many samples in the training set that cannot be represented by the generator network as it is not surjective. Enforcing the mean squared error (MSE) loss for such samples per Eq. (5.8) is not appropriate and leads to blurry reconstructions. Hence, we propose using the following losses for training the encoder.

Approximate Semantic Consistency: To also make sure that our reconstructions are semantically consistent we add:

$$\mathcal{L}_{semantic} = \mathbb{E}_{z \sim \mathcal{N}(0,I)} \Big[\max(\|G(\mathbf{z}) - G(I(G(\mathbf{z})))\|_2^2, \eta) \Big]. \tag{5.9}$$

The use of L_2 norm is not a good distance measure between two images, and minimizing the L_2 distance results in blurry reconstructions. Hence, we use hinge loss on L_2 norm in our formulation. The use of hinge loss in combination with adversarial loss (which we define later) searches for sharp reconstructions that are within η L_2 norm ball of reconstruction error, instead of blurry reconstructions obtained by minimizing just the L_2 reconstruction error.

Latent Code Recovery: In addition to maintaining semantic consistency between the generated images and inverted reconstructions, we recover the latent codes by making sure they are close to the sampled **z**:

$$\mathcal{L}_{latent} = \|z - I(G(z))\|_2^2. \tag{5.10}$$

Inverted Image Distribution Consistency: We want the images that are generated by the inversion $G(I(x))$ to have the same distribution as the images that are generated from samples of the domain space of the generator $G(z)$. Therefore, we add a discriminator at training time for which the the real samples are the generations $G(\mathbf{z})$ and the fake samples are the inversions $G(I(G(z))$:

$$\mathcal{L}_{adv}(I, D) = \mathbb{E}_{z\sim\mathcal{N}(0,I)}\Big[\log(D_{\theta_D}(G(z))) - \log(1 - D_{\theta_D}(G(I(G(z)))))\Big]. \tag{5.11}$$

This adversarial loss is crucial to improving the quality of the reconstructions.

5.4.2 Training

The encoder model is trained using a combination of the three loss terms introduced above. The objective function can be written as

$$\min_{\theta_I}\max_{\theta_D} \lambda_1\mathcal{L}_{adv}(I, D) + \lambda_2\mathcal{L}_{semantic}(I) + \lambda_3\mathcal{L}_{latent}(I).$$

We set $\lambda_2 = 100, \lambda_1 = \lambda_3 = 1$ so that the semantic consistency gets enforced early on in the training, with the other two losses getting minimized gradually to correct for the distribution mismatch. We train in an iterative adversarial fashion to update the parameters of I and D.

5.4.3 Adversarial Defenses

The objective of adversarial defense mechanisms is to make the classifiers robust to any class of adversarial perturbation. In this chapter, we consider norm-bounded perturbations – the most common form of adversarial attacks used till date. However, our framework is general and can be extended to other forms of attacks as well. Given an adversarially perturbed image $x^{adv} = x + \delta$, projection-based defense mechanisms project the adversarial sample x^{adv} to the manifold representing the input dataset. In DefenseGAN [85], the image manifold is first modeled by training a GAN on the input dataset. The perturbed sample x^{adv} is then projected to the generative manifold by solving the optimization (5.5).

In this work, we replace the inference step in DefenseGAN using our proposed InvGAN framework in Algorithm 5.1. Given an input image x, the approximate latent code is first obtained by passing through the inverter network $I(\cdot)$, which can then be used as an initialization to the optimization (5.5).

Attack Detection: In addition to robust classification, we introduce a framework for detecting adversarially perturbed samples. In DefenseGAN, the projection distance in the image space (i.e., $\|x - x^{proj}\|$) is used to determine whether the input

Algorithm 5.1: InvGAN

Input: Image x, GAN G, encoder I, number of random restarts R, number of gradient descent steps T, learning rate η, $\sigma > 0$.

`// Do` R `random restarts.`

for $r \leftarrow 1$ **to** R **do**

  Sample $\tau \sim \mathcal{N}(0, \sigma^2 I)$.

  Set $z_0^{(r)} \leftarrow I(x) + \tau$.

  `// Do` T `steps of gradient descent.`

  for $t \leftarrow 0$ **to** $T - 1$ **do**

    $z_{t+1}^{(r)} \leftarrow z_t^{(r)} - \eta \nabla_{z=z_t^{(r)}} \|G(z) - x\|_2^2$

  end

end

$r^* \leftarrow \text{argmin}_r \|G(z_T^{(r)}) - x\|_2^2$.

$x^{proj} \leftarrow G(z_T^{(r^*)})$.

Output: x^{proj}.

image x is adversarially manipulated. This measure works well when the amount of perturbation is large. For example, in DefenseGAN [85], $\epsilon = 0.3$ in $[0, 1]$ is used for FGSM attack. However, when the perturbation is small (e.g. CW attack), the projection distance may not be a proper measure. Instead, we propose to detect the adversarial images by measuring the *semantic distance* between the input image and the projected image. Specifically, let the trained classification network f be decomposed as $f(x) = C \circ \Phi(x)$ where C denotes the last layer of the network, and Φ denotes all layers except the final layer. $\Phi(x)$ gives a feature representation of the image x. We define attack detection score $\mathcal{A}(x)$ as

$$\mathcal{A}(x) = \|\Phi(x) - \Phi(x^{proj})\|_2. \tag{5.12}$$

If the features extracted from the input image and its projection have a large distance, it means x and x^{proj} are not *consistent* and will be viewed as an adversarial

example.

5.5 Experimental Results

Our proposed approach is evaluated on four datasets: MNIST [56], Fashion-MNIST [112], and CIFAR-10 [54]. We pretrain GANs on all the datasets using the network architectures presented in [70]. We use DCGAN architecture for MNIST and Fasion-MNIST, and GANs with residual blocks for CIFAR-10. The architecture of the inverter I is the mirror image of the generator. The inverter network is trained for 100K iterations using the Adam optimizer with $\beta_1 = 0.5$, and $\beta_2 = 0.999$.

5.5.1 Projecting natural images onto the learned data manifold

In this experiment, we consider the task of inferring the latent representation z of an input image x from learned data manifold G and reconstructing the input by $G(z^*)$. The closeness of the reconstructed image to the input illustrates the strength of the inversion scheme. The following quantitative metrics are considered for evaluation: (1) Inception score (IS) [84], (2) Fréchet inception distance (FID) [37] of the reconstructed samples, (3) MSE between the input and reconstruction. The proposed InvGAN is compared with ALI [19] and AGE [103] on the CIFAR-10 test set. We also consider the following baselines:

- Direct optimization: $z^* = \text{argmin}_z \|G(z) - x\|_2$ is first solved by running gradient descent for 200 iterations. $G(z^*)$ is then treated as the reconstructed image.

- InvGAN with $R = 1, T = 0$: The encoder-decoder scheme similar to ALI and AGE.

- InvGAN with $R = 1, T = 200$: The scheme used to defend against adversarial attacks.

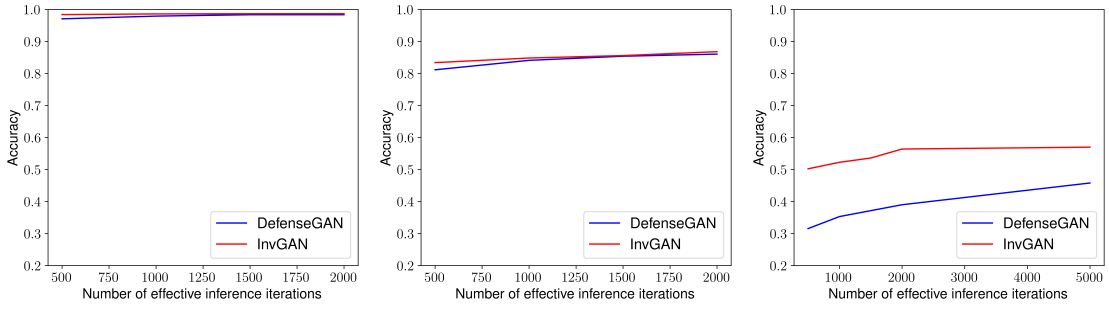
For fair comparisons, in this experiment, we adopt the simple DCGAN architecture without residual blocks in [70] on CIFAR-10. The results are presented in Table 5.1. InvGAN with $T = 0$ achieves the best IS and FID than the competing methods, while direct optimization achieves the best MSE. However, lower MSE does not necessarily produce natural looking images (which is reflected in poorer inception and FID scores) since the MSE loss does not take semantic information into account. Also note that InvGAN only suffers slightly from running several steps of MSE updates. However, these optimization updates offer security against common attacks as will be discussed in the following sections.

Table 5.1: Quantitative evaluation of inference on CIFAR-10 test set.

	MSE	IS	FID
ALI	0.32 ± 0.17	6.12 ± 0.15	57.79
AGE	0.06 ± 0.03	6.43 ± 0.15	39.93
Direct Optimization	$\mathbf{0.03 \pm 0.02}$	6.50 ± 0.20	40.18
InvGAN ($T = 0$)	0.10 ± 0.06	$\mathbf{7.72 \pm 0.16}$	**22.35**
InvGAN ($T = 200$)	0.08 ± 0.04	7.36 ± 0.27	23.91

5.5.2 Running Time Comparisons

Measuring the running time of defense mechanisms is challenging as it depends on the implementation. We propose using the *effective number of gradient descent*



MNIST Fashion-MNIST CIFAR-10

Figure 5.2: Speed - accuracy trade-off curves.

steps as a measure of running time, defined as the product of number of random restarts and the number of iterations per run. We report the classification accuracy of DefenseGAN and InvGAN on MNIST, Fashion-MNIST and CIFAR-10 for clean images by varying the effective number of iterations. The result is presented in Figure 5.2. InvGAN has a significantly shorter run time and offers reconstructions with improved semantic consistency.

5.5.3 Defense Against Adversarial Attacks

In this section, we compare InvGAN with Defense-GAN [85] in defending against common white-box attacks. We consider crafting FGSM [30], PGD and CW [11] attacks on the classifier with different perturbation budget ϵ and feeding the adversarial images to our pipeline. In addition, we also experiment on end-to-end attacks using reparameterization and BPDA [4].

White-box Robust Classification. We use MNIST, Fashion-MNIST, and CIFAR-10 for evaluation. For FGSM and PGD, we select $\epsilon = 25/75$ on MNIST, $\epsilon = 8/25$ on Fashion-MNIST, and $\epsilon = 8/16$ on CIFAR-10. For the CW attack, we

set the binary search step to 4, learning rate to 0.2, and number of iterations to 100. Tables 5.2, 5.3, and 5.4 show the classification accuracy. Compared to DefenseGAN, InvGAN achieves comparable performance on MNIST and Fashion-MNIST, and improved performance on CIFAR-10. In Figure 5.3, we visualize projection results for DefenseGAN and InvGAN when clean images or PGD ($\epsilon = 16$) images are fed as input. We observe that InvGAN reconstructs more shape and color details than DefenseGAN.

Table 5.2: MNIST: Classification accuracy under different white-box attacks. Attack strengths $\epsilon = 25/75$ for RAND, FGSM and PGD.

	Clean	RAND	FGSM	PGD	CW
No Defense	0.99	0.99/0.99	0.89/0.40	0.88/0.04	0.01
DefenseGAN ($R = 10, T = 200$)	0.98	0.98/0.98	0.97/0.84	0.97/0.88	0.97
InvGAN ($R = 1, T = 1000$)	0.99	0.98/0.98	0.97/0.79	0.97/0.81	0.98
InvGAN ($R = 10, T = 200$)	0.99	0.98/0.98	0.97/0.83	0.97/0.87	0.98

Table 5.3: Fashion-MNIST: Classification accuracy under different white-box attacks. Attack strengths $\epsilon = 8/25$ for RAND, FGSM and PGD.

	Clean	RAND	FGSM	PGD	CW
No Defense	0.91	0.91/0.90	0.56/0.24	0.47/0.09	0.06
DefenseGAN ($R = 10, T = 200$)	0.86	0.86/0.86	0.84/0.76	0.84/0.79	0.85
InvGAN ($R = 1, T = 1000$)	0.86	0.86/0.86	0.82/0.70	0.83/0.73	0.85
InvGAN ($R = 10, T = 200$)	0.87	0.86/0.86	0.84/0.76	0.84/0.78	0.86

Table 5.4: CIFAR-10: Classification accuracy under different white-box attacks. Attack strengths $\epsilon = 8/16$ for RAND, FGSM and PGD.

	Clean	RAND	FGSM	PGD	CW
No Defense	0.95	0.92/0.82	0.27/0.19	0.03/0.02	0.03
DefenseGAN ($R = 10, T = 500$)	0.46	0.46/0.46	0.45/0.43	0.44/0.44	0.44
InvGAN ($R = 1, T = 1000$)	0.59	0.58/0.58	0.55/0.50	0.55/0.53	0.55
InvGAN ($R = 10, T = 200$)	0.56	0.56/0.55	0.53/0.47	0.53/0.50	0.53

Figure 5.3: CIFAR-10: Qualitative comparison between DefenseGAN ($R = 10, T = 500$) and InvGAN ($R = 1, T = 1000$).

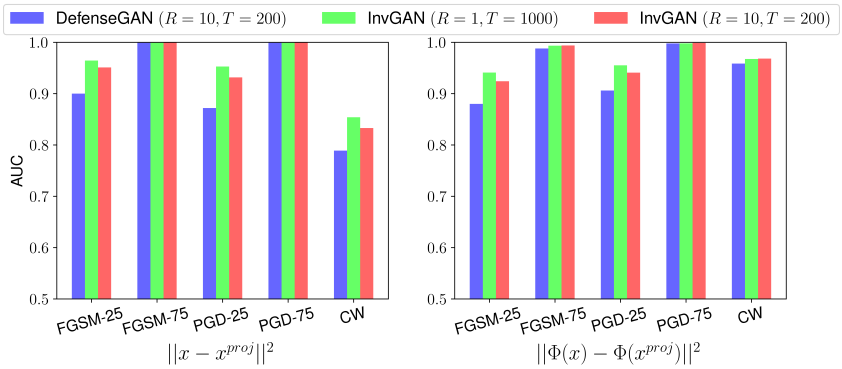


Figure 5.4: MNIST: Attack detection performance (AUC) of DefenseGAN and InvGAN using image (left) and semantic (right) distance.

Attack Detection. In Figures 5.4, 5.5, and 5.6, we compare the area under ROC curve (AUC) scores for attack detection between DefenseGAN and InvGAN on different adversarial attacks. Both image distance [85] and the proposed semantic distance are evaluated. It is clear that semantic distance is more suitable

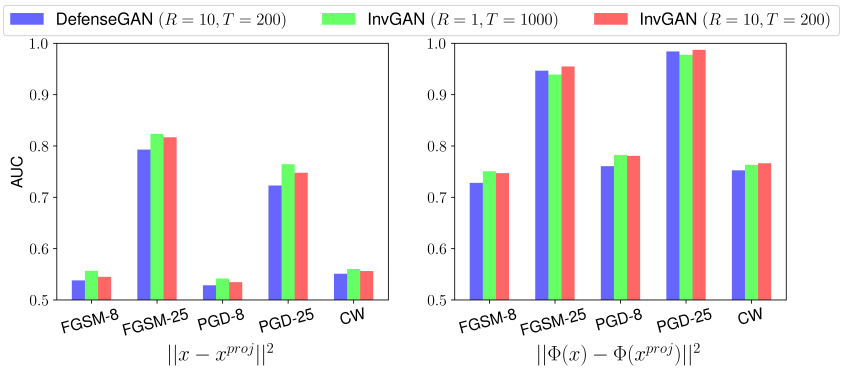


Figure 5.5: Fashion-MNIST: Attack detection performance (AUC) of DefenseGAN and InvGAN using image (left) and semantic (right) distance.

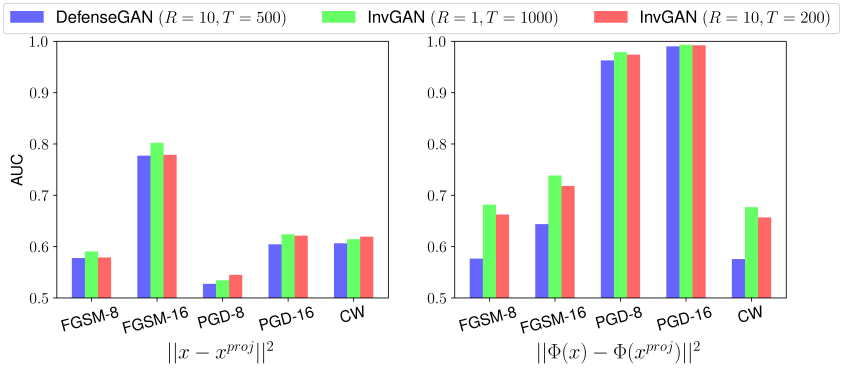


Figure 5.6: CIFAR-10: Attack detection performance (AUC) of DefenseGAN and InvGAN using image (left) and semantic (right) distance.

for attack detection, and InvGAN achieves improved AUC performance than DefenseGAN. Furthermore, comparing the classification accuracy in Tables 5.2-5.4 and attack detection performance in Figures 5.4-5.6, we observe that for large perturbations where the classification accuracy of InvGAN is low, attacks can be detected easily. On the other hand, for minute perturbations, InvGAN successfully removes perturbation and achieves high accuracy, but the attack becomes more challenging to detect. This suggests that our attack detection and robust classification scheme offers orthogonal benefits – when one fails, the other succeeds.

Defense for BPDA Attack. The inability to compute gradients for projection-based defense methods makes crafting effective white-box attacks difficult. BPDA [4]

Table 5.5: Classification accuracy and detection performance under BPDA attack.

	MNIST		Fashion-MNIST		CIFAR-10	
	ACC	DET	ACC	DET	ACC	DET
DefenseGAN ($R = 10, T = 200$)	0.66	0.93	0.59	0.91	0.44	0.54
InvGAN ($R = 1, T = 1000$)	0.44	0.92	0.34	0.87	0.18	0.72
InvGAN ($R = 10, T = 200$)	0.58	0.93	0.42	0.91	0.36	0.68

approximates the gradients by the straight through estimator (STE). That is, during backpropagation, the projection operation is replaced by the identity function. Adversarial attacks are then crafted by minimizing the CW loss. Using the strategy, Athalye *et al.* [4] brought the classification accuracy on MNIST to 55% in their setting. We present classification accuracy and detection performance in Table 5.5. Notice that DefenseGAN achieves higher accuracy than InvGAN under BPDA attack, which is mainly due to the imperfection of DefenseGAN reconstruction. The phenomenon will be more apparent in the next section when adversarial attacks are guaranteed to be on the manifold.

Defense for Overpowered Attack. To break DefenseGAN, the adversarial images have to be those x^{adv} whose projection $x^{proj} = G(z^*)$ is also adversarial to the classifier. One possible approach is to directly assign $x^{adv} \leftarrow G(z^*)$, where

$$z^* = \underset{z}{\operatorname{argmin}} \|x - G(z)\|_2^2 + l_{CW}(G(z), y). \tag{5.13}$$

If the projection operation in DefenseGAN is *perfect*, we will have $x^{proj} = x^{adv} = G(z^*)$, which will fool the classifier. However, Athalye *et al.* [4] empirically found out such approach is not successful due to the imperfect projection of DefenseGAN.

To improve the attack, Jalal *et al.* [42] proposed to solve for (5.4), which finds $G(z^*)$ such that images within a small neighborhood can fool the classifier in expectation. We craft adversarial examples using Overpowered Attack with the same setting as in [42]. We present classification accuracy and detection performance in Table 5.6. Notice that under the Overpowered Attack, DefenseGAN outperforms InvGAN in classification accuracy. However, we found out that with the perturbation budget 0.0051 (per-pixel L2 distance) used by Jalal *et al.* [42], Overpowered Attack generates *perceptible* perturbations to the input images. In Figure 5.7, it is clear that Overpowered Attack adds/removes small strokes to the MNIST images, which drastically changes the underlying semantics. For example, the clean image of digit '6' becomes a digit '8'. We argue that such perturbations are *not valid attacks* since there is no reason to expect a classifier to predict a '6' when the image is visually an '8'. Furthermore, if we reduce the perturbation budget, the optimization (5.4) may not converge to a feasible solution and oscillates between minimizing the L2 loss and the CW loss. From the images highlighted in green boxes in Figure 5.7, we also observe that the imperfect projection of DefenseGAN leads to some failures for the attack, and hence higher classification accuracy.

Table 5.6: Classification accuracy and detection performance under BPDA attack. *Value rounded from 0.0041.

	MNIST		Fashion-MNIST	
	ACC	DET	ACC	DET
No Defense	0.00*	-	0.01	-
DefenseGAN ($R = 10, T = 200$)	0.20	0.41	0.36	0.35
InvGAN ($R = 1, T = 1000$)	0.13	0.34	0.24	0.26
InvGAN ($R = 10, T = 200$)	0.11	0.25	0.32	0.32

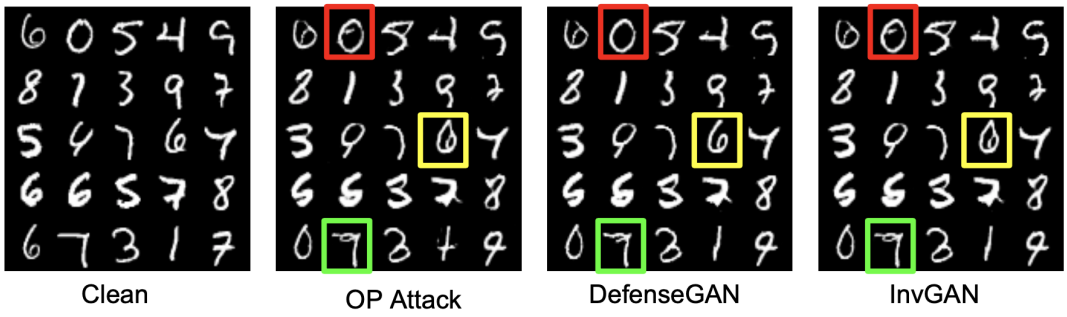


Figure 5.7: Visualization of Overpowered Attack and reconstructions by DefenseGAN ($R = 10, T = 200$) and InvGAN ($R = 1, T = 1000$).

5.5.4 Ablation study

In this section, we study the effect of disabling the adversarial loss. More specifically, we train the encoder model by setting $\eta = 0$, $\lambda_1 = 0$, $\lambda_2 = 1$, and $\lambda_3 = 0$. As can be seen from Table 5.7, although this network achieves a lower MSE, the images are perceptually less realistic, and is reflected in lower classification accuracy.

Table 5.7: Analyzing the effect of adversarial loss in InvGAN.

	MSE	IS	FID	Accuracy
w/o $\mathcal{L}_{adv}$	0.08 ± 0.04	7.54 ± 0.17	28.07	0.603
Ours	0.10 ± 0.06	$\mathbf{7.72 \pm 0.16}$	**19.85**	**0.625**

5.6 Conclusion

In this chapter, we introduce InvGAN – a novel data-free and model-based inversion framework for solving the inference problem in GANs. Our approach involves training an encoder function capable of inverting the generator network back to the latent space. The encoder function is trained using a novel loss function

that achieves superior inversion results compared to the contemporary methods performing inference. The usefulness of our inversion scheme is demonstrated for the problem of adversarial defenses, where our inversion scheme has been shown to achieve dramatic improvements in defense performance, running time and attack detection over DefenseGAN. Finally, we study attack methods that claim to break DefenseGAN and empirically found out that in the case of Overpowered Attack, perceptible changes are made to the clean images, which violates the definition of adversarial perturbation. We propose that on-manifold robustness is an interesting topic and should be carefully evaluated.

Chapter 6: Conclusions and Future Research Directions

6.1 Conclusions

In this dissertation, we begin with an overview of existing priors and constraints posed *implicitly* by common neural network architecture designs. Despite the success of these implicit priors and constraints, which require minimal adhoc feature engineering and achieve good performance, DNNs are shown to be vulnerable under distribution shift and adversarial perturbation. This motivates us to develop task-dependent constraints and priors.

In Chapters 2 and 3, we showed the benefit of combining the neighborhood structure of deep features for face subject clustering. In Chapter 2, we proposed an exemplar-based method to improve the pairwise similarity measure for clustering. In addition to surveillance application, we also demonstrated how the proposed algorithm can be applied to automatically curate datasets with noisy labels, which significantly reduces the cost of data annotation. In Chapter 3, we utilized the local density in the feature space to construct the pairwise similarity measure. Feature points that are within the high density regions with each other have higher similarity than those that are not. Experimental evaluations showed that the density-based approach achieves even better performance than the exemplar-based approach.

In Chapter 4, we showed that the imaging geometry of computed tomography can be used as a data consistency constraint during training. The network learns to reduce structured artifacts both in the projection (sinogram) and image domains. Extensive experiments are conducted on a large-scale simulated dataset and a real clinical dataset. The proposed DuDoNet outperforms competing methods by a large margin.

In Chapter 5, we proposed an inversion method for a pretrained GAN and improved the efficiency of DefenseGAN for defending against common norm-bounded attacks. The inversion model was used as a prior when inferring the latent codes for input images. Therefore, semantic information in the input image can be reconstructed using a smaller number of gradient descent steps. We also developed an algorithm for detecting adversarial examples based on feature distance. The adversarial detection was shown to bring orthogonal benefits to perturbation removal for secured classification. Finally, we investigate attacks that claim to break DefenseGAN. We found out that the method which crafts on-manifold adversarial examples are the most effective one. However, existing works tend to craft images with drastically different semantics from the source images, which we view as invalid. Future studies for valid on-manifold adversarial examples and on-manifold robustness could shed light on how to further improve projection-based defense methods.

6.2 Future Research Directions

In Chapters 2 and 3, the framework of subject clustering includes feature extraction using a pretrained DNN followed by hierarchical clustering. In order to train the DNN, a small-size annotated face identification dataset is required. In general applications, annotated datasets may not be available. Therefore, one possible research direction is to develop an unsupervised representation learning algorithm using a large-scale image database, and then apply metric learning techniques such that visually similar faces have smaller distance in the feature space. In Chapter 2, we demonstrated that clustering algorithm can be used to automatically curate datasets with noisy labels, and DNN models can be improved by fine-tuning on the curated dataset. A possible research direction is to apply the procedure iteratively, and investigate whether a powerful DNN can be trained with minimum efforts in human annotation.

In Chapter 4, we showed that dual domain learning on simulated data improves the performance and generalization ability of CNNs for metal artifact reduction. Although promising results are demonstrated in clinical CT images, it is still unclear whether this approach can be generalized to CT images taken by machines with significantly different X-ray spectrum and imaging geometry. One research direction is to investigate semi-supervised learning for metal artifact reduction. In addition to paired simulated data, we could utilize a large number of CT images with or without metal artifact to improve the robustness of CNNs.

In Chapter 5, we empirically showed that out-of-manifold adversarial pertur-

bations can be removed or detected by projecting onto the learned data manifold. To break projection-based defenses (e.g. DefenseGAN and the proposed InvGAN), crafting on-manifold adversarial examples is necessary. Therefore, one promising research direction is to quantitatively study and improve the on-manifold robustness of classifiers. Specifically, given a trained classifier, we would like to ask (1) how to craft *valid* on-manifold adversarial examples effectively, and (2) can we use these adversarial examples to improve the robustness of the classifier?

Bibliography

[1] Cycada: Cycle consistent adversarial domain adaptation. In *International Conference on Machine Learning (ICML)*, 2018.

[2] J. Adler and O. Öktem. Learned primal-dual reconstruction. *IEEE Transactions on Medical Imaging*, 37(6):1322–1332, June 2018.

[3] Enrique Amigó, Julio Gonzalo, Javier Artiles, and Felisa Verdejo. A comparison of extrinsic clustering evaluation metrics based on formal constraints. *Information retrieval*, 12(4):461–486, 2009.

[4] Anish Athalye, Nicholas Carlini, and David Wagner. Obfuscated gradients give a false sense of security: Circumventing defenses to adversarial examples. In *International Conference on Machine Learning (ICML)*, pages 274–283, 2018.

[5] David Bau, Hendrik Strobelt, William Peebles, Jonas Wulff, Bolei Zhou, Jun-Yan Zhu, and Antonio Torralba. Semantic photo manipulation with a generative image prior. *ACM Transactions on Graphics*, 38:59:1–59:11, 2019.

[6] Beer. Bestimmung der absorption des rothen lichts in farbigen flüssigkeiten. *Annalen der Physik und Chemie*, 162(5):78–88, 1852.

[7] Ashish Bora, Ajil Jalal, Eric Price, and Alexandros G Dimakis. Compressed sensing using generative models. In *International Conference on Machine Learning (ICML)*, pages 537–546, 2017.

[8] Konstantinos Bousmalis, Nathan Silberman, David Dohan, Dumitru Erhan, and Dilip Krishnan. Unsupervised pixel-level domain adaptation with generative adversarial networks. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, July 2017.

[9] Andrew Brock, Jeff Donahue, and Karen Simonyan. Large scale GAN training for high fidelity natural image synthesis. *CoRR*, abs/1809.11096, 2018.

[10] Pau Panareda Busto and Juergen Gall. Open set domain adaptation. In *IEEE International Conference on Computer Vision (ICCV)*, 2017.

[11] Nicholas Carlini and David Wagner. Towards evaluating the robustness of neural networks. In *2017 IEEE Symposium on Security and Privacy (SP)*, pages 39–57. IEEE, 2017.

[12] Wei-Cheng Chang, Ching-Pei Lee, and Chih-Jen Lin. A revisit to support vector data description (SVDD), 2013.

[13] J.-C. Chen, V.M. Patel, and Rama Chellappa. Unconstrainded face verification using deep CNN features. In *IEEE Winter Conference on Applications of Computer Vision (WACV)*, 2016.

[14] Antonia Creswell and Anil Anthony Bharath. Inverting the generator of a generative adversarial network. *IEEE Transactions on Neural Networks and Learning Systems*, 2018.

[15] Kostadin Dabov, Alessandro Foi, Vladimir Katkovnik, and Karen Egiazarian. Image denoising by sparse 3-D transform-domain collaborative filtering. *IEEE Transactions on Image Processing*, 16(8):2080–2095, 2007.

[16] Jeff Donahue, Philipp Krähenbühl, and Trevor Darrell. Adversarial feature learning. *arXiv preprint arXiv:1605.09782*, 2016.

[17] Alexey Dosovitskiy and Thomas Brox. Generating images with perceptual similarity metrics based on deep networks. In *Advances in Neural Information Processing Systems*, 2016.

[18] Xinhui Duan, Li Zhang, Yongshun Xiao, Jianping Cheng, Zhiqiang Chen, and Yuxiang Xing. Metal artifact reduction in CT images by sinogram TV inpainting. In *Nuclear Science Symposium Conference Record, 2008. NSS'08. IEEE*, pages 4175–4177. IEEE, 2008.

[19] Vincent Dumoulin, Mohamed Ishmael Diwan Belghazi, Ben Poole, Alex Lamb, Martin Arjovsky, Olivier Mastropietro, and Aaron Courville. Adversarially learned inference. 2017.

[20] E. Elhamifar and R. Vidal. Sparse subspace clustering. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2009.

[21] E. Elhamifar and R. Vidal. Sparse subspace clustering: Algorithm, theory, and applications. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2013.

[22] Martin Ester, Hans-Peter Kriegel, Jörg Sander, and Xiaowei Xu. A density-based algorithm for discovering clusters a density-based algorithm for discovering clusters in large spatial databases with noise. In *Proceedings of the Second International Conference on Knowledge Discovery and Data Mining*, KDD, pages 226–231, 1996.

[23] Rong-En Fan, Kai-Wei Chang, Cho-Jui Hsieh, Xiang-Rui Wang, and Chih-Jen Lin. LIBLINEAR: A library for large linear classification. *Journal of Machine Learning Research*, 9:1871–1874, 2008.

[24] Brendan J. Frey and Delbert Dueck. Clustering by passing messages between data points. *Science*, 315, 2007.

[25] Yaroslav Ganin and Victor Lempitsky. Unsupervised domain adaptation by backpropagation. In *IEEE International Conference on Machine Learning (ICML)*, 2015.

[26] A.S. Georghiades, P.N. Belhumeur, and D.J. Kriegman. From few to many: Illumination cone models for face recognition under variable lighting and pose. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 23(6):643–660, 2001.

[27] Kamran Ghasedi Dizaji, Amirhossein Herandi, Cheng Deng, Weidong Cai, and Heng Huang. Deep clustering via joint convolutional autoencoder embedding and relative entropy minimization. In *IEEE International Conference on Computer Vision (ICCV)*, Oct 2017.

[28] L. Gjesteby, Q. Yang, Y. Xi, Y. Zhou, J. Zhang, and G. Wang. Deep learning methods to guide CT image reconstruction and reduce metal artifacts. In *SPIE Medical Imaging*, 2017.

[29] Lars Gjesteby, Qingsong Yang, Yan Xi, Bernhard Claus, Yannan Jin, Bruno De Man, and Ge Wang. Reducing metal streak artifacts in CT images via deep learning: Pilot results. In *The 14th International Meeting on Fully Three-Dimensional Image Reconstruction in Radiology and Nuclear Medicine*, pages 611–614, 2017.

[30] Ian J. Goodfellow, Jonathon Shlens, and Christian Szegedy. Explaining and harnessing adversarial examples. In *International Conference on Learning Representations*, 2015.

[31] K. Chidananda Gowda and G. Krishna. Agglomerative clustering using the concept of mutual nearest neighbourhood. *Pattern Recognition*, 10(2):105–112, 1978.

[32] Jinjin Gu, Yujun Shen, and Bolei Zhou. Image processing using multi-code GAN prior. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020.

[33] Jun Guo and Hongyang Chao. Building dual-domain representations for compression artifacts reduction. In *European Conference on Computer Vision (ECCV)*, pages 628–644. Springer, 2016.

[34] Yandong Guo, Lei Zhang, Yuxiao Hu, Xiaodong He, and Jianfeng Gao. MS-Celeb-1M: A dataset and benchmark for large scale face recognition. In *European Conference on Computer Vision (ECCV)*. Springer, 2016.

[35] H. Gupta, K. H. Jin, H. Q. Nguyen, M. T. McCann, and M. Unser. Iterative metal artifact reduction for x-ray computed tomography using unmatched projector/backprojector pairs. *IEEE Transactions on Medical Imaging*, 43(6):3019–3033, 2016.

[36] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 770–778, 2016.

[37] Martin Heusel, Hubert Ramsauer, Thomas Unterthiner, Bernhard Nessler, and Sepp Hochreiter. GANs trained by a two time-scale update rule converge to a local nash equilibrium. In *Advances in Neural Information Processing Systems*, 2017.

[38] G. B. Huang, M. Mattar, T. Berg, and E. Learned-Miller. Labeled faces in the wild: A database for studying face recognition in unconstrained environments. In *Workshop on Faces in Real-Life Images: Detection, Alignment, and Recognition*, 2008.

[39] Gao Huang, Zhuang Liu, Laurens van der Maaten, and Kilian Q Weinberger. Densely connected convolutional networks. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017.

[40] Andrew Ilyas, Shibani Santurkar, Dimitris Tsipras, Logan Engstrom, Brandon Tran, and Aleksander Madry. Adversarial examples are not bugs, they are features. In *Advances in Neural Information Processing Systems*. 2019.

[41] Phillip Isola, Jun-Yan Zhu, Tinghui Zhou, and Alexei A Efros. Image-to-image translation with conditional adversarial networks. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1125–1134, 2017.

[42] Ajil Jalal, Andrew Ilyas, Constantinos Daskalakis, and Alexandros G. Dimakis. The robust manifold defense: Adversarial training using generative models. *CoRR*, abs/1712.09196, 2017.

[43] Pan Ji, Tong Zhang, Hongdong Li, Mathieu Salzmann, and Ian Reid. Deep subspace clustering network. In *Advances in Neural Information Processing Systems*, 2017.

[44] K. H. Jin, M. T. McCann, E. Froustey, and M. Unser. Deep convolutional neural network for inverse problems in imaging. *IEEE Transactions on Image Processing*, 26(9):4509–4522, Sept 2017.

[45] SouYoung Jin, Hang Su, Chris Stauffer, and Erik Learned-Miller. End-to-end face detection and cast grouping in movies using erdos-renyi clustering. In *IEEE International Conference on Computer Vision (ICCV)*, Oct 2017.

[46] Justin Johnson, Alexandre Alahi, and Li Fei-Fei. Perceptual losses for real-time style transfer and super-resolution. In *European Conference on Computer Vision (ECCV)*, 2016.

[47] Avinash C. Kak and Malcolm Slaney. *Principles of computerized tomographic imaging*. Society for Industrial and Applied Mathematics, 2001.

[48] Willi A Kalender, Robert Hebel, and Johannes Ebersberger. Reduction of CT artifacts caused by metallic implants. *Radiology*, 164(2):576–577, 1987.

[49] Tero Karras, Timo Aila, Samuli Laine, and Jaakko Lehtinen. Progressive growing of GANs for improved quality, stability, and variation. In *International Conference on Learning Representations*, 2018.

[50] Tero Karras, Samuli Laine, and Timo Aila. A style-based generator architecture for generative adversarial networks. *CoRR*, abs/1812.04948, 2018.

[51] Diederik Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

[52] B. F. Klare, B. Klein, E. Taborsky, A. Blanton, J. Cheney, K. Allen, P. Grother, A. Mah, M. Burge, and A. K. Jain. Pushing the frontiers of unconstrained face detection and recognition: IARPA JANUS Benchmark A. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015.

[53] A. Krizhevsky, I. Sutskever, and G. E. Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in Neural Information Processing Systems*, pages 1097–1105, 2012.

[54] Alex Krizhevsky, Vinod Nair, and Geoffrey Hinton. CIFAR-10 (canadian institute for advanced research).

[55] Takio Kurita. An efficient agglomerative clustering algorithm using a heap. *Pattern Recognition*, 24(3):205–209, 1991.

[56] Yann Lecun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. Gradient-based learning applied to document recognition. In *Proceedings of the IEEE*, pages 2278–2324, 1998.

[57] Yann LeCun, Corinna Cortes, and Christopher JC Burges. The MNIST database of handwritten digits, 1998.

[58] Christian Ledig, Lucas Theis, Ferenc Huszár, Jose Caballero, Andrew Cunningham, Alejandro Acosta, Andrew P Aitken, Alykhan Tejani, Johannes Totz, Zehan Wang, et al. Photo-realistic single image super-resolution using a generative adversarial network. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, volume 2, page 4, 2017.

[59] Jaakko Lehtinen, Jacob Munkberg, Jon Hasselgren, Samuli Laine, Tero Karras, Miika Aittala, and Timo Aila. Noise2Noise: Learning image restoration without clean data. In *International Conference on Machine Learning (ICML)*, volume 80, pages 2965–2974, 2018.

[60] Wei-An Lin, Jun-Cheng Chen, and Rama Chellappa. A proximity-aware hierarchical clustering of faces. In *IEEE Conference on Automatic Face and Gesture Recognition (FG)*, 2017.

[61] Zachary C Lipton and Subarna Tripathi. Precise recovery of latent vectors from generative adversarial networks. *International Conference on Learning Representation (Workshops)*, 2017.

[62] Guangcan Liu and Shuicheng Yan. Latent low-rank representation for subspace segmentation and feature extraction. In *IEEE International Conference on Computer Vision (ICCV)*, pages 1615–1622, 2011.

[63] Weiyang Liu, Yandong Wen, Zhiding Yu, Ming Li, Bhiksha Raj, and Le Song. Sphereface: Deep hypersphere embedding for face recognition. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, July 2017.

[64] S. Lloyd. Least squares quantization in PCM. *IEEE Transactions on Information Theory*, 28(2):129–137, 1982.

[65] M. Long, H. Zhu, J. Wang, and M. I. Jordan. Deep transfer learning with joint adaptation networks. In *IEEE International Conference on Machine Learning (ICML)*, 2017.

[66] Markku Makitalo and Alessandro Foi. Optimal inversion of the anscombe transformation in low-count Poisson image denoising. *IEEE Transactions on Image Processing*, 20(1):99–109, 2011.

[67] Abolfazl Mehranian, Mohammad Reza Ay, Arman Rahmim, and Habib Zaidi. X-ray CT metal artifact reduction using wavelet domain L0 sparse regularization. *IEEE Transactions on Medical Imaging*, 32:1707–1722, 2013.

[68] Dongyu Meng and Hao Chen. Magnet: A two-pronged defense against adversarial examples. In *Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security*, CCS '17, pages 135–147. ACM, 2017.

[69] Esther Meyer, Rainer Raupach, Michael Lell, Bernhard Schmidt, and Marc Kachelrieß. Normalized metal artifact reduction (NMAR) in computed tomography. *Medical physics*, 37(10):5482–5493, 2010.

[70] Takeru Miyato, Toshiki Kataoka, Masanori Koyama, and Yuichi Yoshida. Spectral normalization for generative adversarial networks. In *International Conference on Learning Representations*, 2018.

[71] Seyed-Mohsen Moosavi-Dezfooli, Alhussein Fawzi, Omar Fawzi, and Pascal Frossard. Universal adversarial perturbations. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 86–94. IEEE Computer Society, 2017.

[72] Andrew Y. Ng, Michael I. Jordan, and Yair Weiss. On spectral clustering: Analysis and an algorithm. In *Advances in Neural Information Processing Systems*, pages 849–856, 2001.

[73] Charles Otto, Dayong Wang, and Anil K. Jain. Clustering millions of faces by identity. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2017.

[74] J. Pan, W. Ren, Z. Hu, and M. Yang. Learning to deblur images with exemplars. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2018.

[75] Dohyung Park, Constantine Caramanis, and Sujay Sanghavi. Greedy subspace clustering. In *Neural Information Processing Systems*, December 2014.

[76] Hyung Suk Park, Yong Eun Chung, Sung Min Lee, Hwa Pyung Kim, and Jin Keun Seo. Sinogram-consistency learning in CT for metal artifact reduction. *arXiv preprint arXiv:1708.00607*, 2017.

[77] Adam Paszke, Sam Gross, Soumith Chintala, Gregory Chanan, Edward Yang, Zachary DeVito, Zeming Lin, Alban Desmaison, Luca Antiga, and Adam Lerer. Automatic differentiation in PyTorch. In *NIPS-W*, 2017.

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

[79] Vishal M. Patel, Hien Van Nguyen, and Rene Vidal. Latent space sparse subspace clustering. In *IEEE International Conference on Computer Vision (ICCV)*, December 2013.

[80] Chong Peng, Zhao Kang, and Qiang Cheng. Subspace clustering via variance regularized ridge regression. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, July 2017.

[81] Rajeev Ranjan, Carlos D Castillo, and Rama Chellappa. L2-constrained softmax loss for discriminative face verification. *arXiv preprint arXiv:1703.09507*, 2017.

[82] Rajeev Ranjan, Swami Sankaranarayanan, Carlos D. Castillo, and Rama Chellappa. An all-in-one convolutional neural network for face analysis. In *IEEE International Conference on Automatic Face Gesture Recognition (FG)*, pages 17–24, 2017.

[83] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-Net: Convolutional networks for biomedical image segmentation. In *Medical Image Computing and Computer Assisted Intervention (MICCAI)*, pages 234–241. Springer, 2015.

[84] Tim Salimans, Ian J. Goodfellow, Wojciech Zaremba, Vicki Cheung, Alec Radford, and Xi Chen. Improved techniques for training GANs. In *Advances in Neural Information Processing Systems*, 2016.

[85] Pouya Samangouei, Maya Kabkab, and Rama Chellappa. Defense-gan: Protecting classifiers against adversarial attacks using generative models. *International Conference on Learning Representation*, 2018.

[86] Swami Sankaranarayanan, Yogesh Balaji, Carlos D. Castillo, and Rama Chellappa. Generate to adapt: Aligning domains using generative adversarial networks. *CoRR*, abs/1704.01705, 2017.

[87] Thomas Schlegl, Philipp Seeböck, Sebastian M. Waldstein, Ursula Schmidt-Erfurth, and Georg Langs. Unsupervised anomaly detection with generative adversarial networks to guide marker discovery. In *Information Processing in Medical Imaging - 25th International Conference, IPMI 2017, Boone, NC, USA, June 25-30, 2017, Proceedings*, 2017.

[88] Bernhard Schölkopf, John C. Platt, John Shawe-Taylor, Alex J. Smola, and Robert C. Williamson. Estimating the support of a high-dimensional distribution. *Neural Computation*, 13(7), July 2001.

[89] Bernhard Schölkopf, Alex J. Smola, Robert C. Williamson, and Peter L. Bartlett. New support vector algorithms. *Neural Computation*, 12(5):1207–1245, May 2000.

[90] Bernhard Schölkopf, Robert C Williamson, Alex J. Smola, John Shawe-Taylor, and John C. Platt. Support vector method for novelty detection. In *Advances in Neural Information Processing Systems*, pages 582–588. 2000.

[91] Florian Schroff, Dmitry Kalenichenko, and James Philbin. Facenet: A unified embedding for face recognition and clustering. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2015.

[92] S. Sengupta, J.-C. Chen, C. Castillo, V. M. Patel, R. Chellappa, and D. W. Jacobs. Frontal to profile face verification in the wild. In *IEEE Winter Conference on Applications of Computer Vision (WACV)*, pages 1–9. IEEE, 2016.

[93] Jianbo Shi and Jitendra Malik. Normalized cuts and image segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(8):888–905, 2000.

[94] Yichun Shi, Charles Otto, and Anil K. Jain. Face clustering: Representation and pairwise constraints. *CoRR*, abs/1706.05067, 2017.

[95] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014.

[96] Yang Song, Taesup Kim, Sebastian Nowozin, Stefano Ermon, and Nate Kushman. Pixeldefend: Leveraging generative models to understand and defend against adversarial examples. In *International Conference on Learning Representations*, 2018.

[97] Christian Szegedy, Wojciech Zaremba, Ilya Sutskever, Joan Bruna, Dumitru Erhan, Ian Goodfellow, and Rob Fergus. Intriguing properties of neural networks. In *International Conference on Learning Representations*, 2014.

[98] Yaniv Taigman, Ming Yang, Marc'Aurelio Ranzato, and Lior Wolf. Deepface: Closing the gap to human-level performance in face verification. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2014.

[99] David M. J. Tax and Robert P. W. Duin. Support vector domain description. *Pattern Recognition Letters*, 20:1191–1199, 1999.

[100] David M.J Tax and Robert P.W Duin. Support vector data description. *Machine Learning*, 54(1):45–66, 2004.

[101] Eric Tzeng, Judy Hoffman, Kate Saenko, and Trevor Darrell. Adversarial discriminative domain adaptation. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017.

[102] Dmitry Ulyanov, Andrea Vedaldi, and Victor Lempitsky. Deep image prior. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 9446–9454, 2018.

[103] Dmitry Ulyanov, Andrea Vedaldi, and Victor S. Lempitsky. It takes (only) two: Adversarial generator-encoder networks. In *AAAI*. AAAI Press, 2018.

[104] R. Vidal and P. Favaro. Low rank subspace clustering (lrsc). *Pattern Recognition Letters*, 43:47–61, 2014.

[105] Martin J. Wainwright. *High-Dimensional Statistics: A Non-Asymptotic Viewpoint*. Cambridge Series in Statistical and Probabilistic Mathematics. Cambridge University Press, 2019.

[106] Jianing Wang, Yiyuan Zhao, Jack H. Noble, and Benoit M. Dawant. Conditional generative adversarial networks for metal artifact reduction in CT images of the ear. In *Medical Image Computing and Computer Assisted Intervention (MICCAI)*, 2018.

[107] Xintao Wang, Ke Yu, Shixiang Wu, Jinjin Gu, Yihao Liu, Chao Dong, Yu Qiao, and Chen Change Loy. ESRGAN: Enhanced super-resolution generative adversarial networks. In *European Conference on Computer Vision Workshops (ECCVW)*, September 2018.

[108] Cameron Whitelam, Emma Taborsky, Austin Blanton, Brianna Maze, Jocelyn Adams, Tim Miller, Nathan Kalka, Anil K. Jain, James A. Duncan, Kristen Allen, Jordan Cheney, and Patrick Grother. IARPA JANUS Benchmark B face dataset. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, July 2017.

[109] L. Wolf, T. Hassner, and I. Maoz. Face recognition in unconstrained videos with matched background similarity. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2011.

[110] Lior Wolf, Tal Hassner, and Yaniv Taigman. The one-shot similarity kernel. In *IEEE International Conference on Computer Vision (ICCV)*, 2009.

[111] Tobias Würfl, Florin C. Ghesu, Vincent Christlein, and Andreas Maier. Deep learning computed tomography. In *Medical Image Computing and Computer Assisted Intervention (MICCAI)*, pages 432–440. Springer International Publishing, 2016.

[112] Han Xiao, Kashif Rasul, and Roland Vollgraf. Fashion-MNIST: a novel image dataset for benchmarking machine learning algorithms. 2017.

[113] Junyuan Xie, Ross Girshick, and Ali Farhadi. Unsupervised deep embedding for clustering analysis. In *IEEE International Conference on Machine Learning (ICML)*, 2016.

[114] Liyang Xie, Kaixiang Lin, Shu Wang, Fei Wang, and Jiayu Zhou. Differentially private generative adversarial network. *CoRR*, abs/1802.06739, 2018.

[115] Shiyu Xu and Hao Dang. Deep residual learning enabled metal artifact reduction in CT. In *Medical Imaging 2018: Physics of Medical Imaging*, volume 10573, page 105733O. International Society for Optics and Photonics, 2018.

[116] Ke Yan, Xiaosong Wang, Le Lu, Ling Zhang, Adam P. Harrison, Mohammadhadi Bagheri, and Ronald M. Summers. Deep lesion graphs in the wild: Relationship learning and organization of significant radiology image findings in a diverse large-scale lesion database. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2018.

[117] Bo Yang, Xiao Fu, Nicholas D. Sidiropoulos, and Mingyi Hong. Towards k-means-friendly spaces: Simultaneous deep learning and clustering. In *IEEE International Conference on Machine Learning (ICML)*, 2017.

[118] Jianwei Yang, Devi Parikh, and Dhruv Batra. Joint unsupervised learning of deep representations and image clusters. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.

[119] D. Yi, Z. Lei, S. Liao, and S. Z. Li. Learning face representation from scratch. *arXiv preprint arXiv:1411.7923*, 2014.

[120] Chong You, Daniel Robinson, and Rene Vidal. Scalable sparse subspace clustering by orthogonal matching pursuit. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2016.

[121] Lihi Zelnik-manor and Pietro Perona. Self-tuning spectral clustering. In *Advances in Neural Information Processing Systems*, pages 1601–1608, 2004.

[122] Haimiao Zhang, Bin Dong, and Baodong Liu. A reweighted joint spatial-radon domain CT image reconstruction model for metal artifact reduction. *SIAM J. Imaging Sciences*, 11:707–733, 2018.

[123] He Zhang and Vishal M Patel. Densely connected pyramid dehazing network. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018.

[124] Richard Zhang, Phillip Isola, Alexei A Efros, Eli Shechtman, and Oliver Wang. The unreasonable effectiveness of deep features as a perceptual metric. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018.

[125] X. Zhang, W. Yang, Y. Hu, and J. Liu. DMCNN: Dual-domain multi-scale convolutional neural network for compression artifacts removal. In *IEEE International Conference on Image Processing (ICIP)*, 2018.

[126] Yanbo Zhang and Hengyong Yu. Convolutional neural network-based metal artifact reduction in x-ray computed tomography. *IEEE Transactions on Medical Imaging*, 2018.

[127] Yulun Zhang, Yapeng Tian, Yu Kong, Bineng Zhong, and Yun Fu. Residual dense network for image super-resolution. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018.

[128] Z. Zhang, X. Liang, X. Dong, Y. Xie, and G. Cao. A sparse-view CT reconstruction method based on combination of densenet and deconvolution. *IEEE Transactions on Medical Imaging*, 37(6):1407–1417, June 2018.

[129] Z. Zhang, P. Luo, C. C. Loy, and X. Tang. Joint face representation adaptation and clustering in videos. In *European Conference on Computer Vision (ECCV)*, 2016.

[130] J. Zheng, J.-C. Chen, N. Bodla, V. M. Patel, and R. Chellappa. VLAD-encoded deep convolutional features for unconstrained face verification. In *IEEE International Conference on Pattern Recognition*, 2016.

[131] Zhisheng Zhong, Tiancheng Shen, Yibo Yang, Zhouchen Lin, and Chao Zhang. Joint sub-bands learning with clique structures for wavelet domain super-resolution. *arXiv preprint arXiv:1809.04508*, 2018.

[132] C. Zhu, F. Wen, and J. Sun. A rank-order distance based clustering algorithm for face tagging. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 481–488, 2011.