3-23-2020

Machine Vision, Not Human Vision, Guided Compression Towards Low-Latency and Robust Deep Learning Systems

Zihao Liu
Florida International University, zliu021@fiu.edu

FLORIDA INTERNATIONAL UNIVERSITY

Miami, Florida

MACHINE VISION, NOT HUMAN VISION, GUIDED COMPRESSION TOWARDS LOW-LATENCY AND ROBUST DEEP LEARNING SYSTEMS

A dissertation submitted in partial fulfillment of the

requirements for the degree of

DOCTOR OF PHILOSOPHY

in

ELECTRICAL AND COMPUTER ENGINEERING

by

Zihao Liu

2020

To: Dean John L. Volakis
Department of Electrical & Computer Engineering

This dissertation, written by Zihao Liu, and entitled Machine Vision, NOT Human Vision, Guided Compression towards Low-Latency and Robust Deep Learning Systems, having been approved in respect to style and intellectual content, is referred to you for judgment.

We have read this dissertation and recommend that it be approved.

Nezih Pala

Gang Quan

Jean H. Andrian

Deng Pan

Wujie Wen, Major Professor

Date of Defense: March 23, 2020

The dissertation of Zihao Liu is approved.

Dean John L. Volakis
Department of Electrical & Computer Engineering

Andrés G. Gil
Vice President for Research and Economic Development and Dean of the University Graduate School

Florida International University, 2020

DEDICATION

I dedicate my dissertation work to my family and many friends. A special feeling of gratitude to my loving parents, whose words of encouragement and push for tenacity ring in my ears. My cousin and uncles have never left my side and are very special. I also dedicate this dissertation to my many friends who have supported me throughout the process.

ACKNOWLEDGMENTS

First and foremost, I am grateful to my major advisor, Dr. Wujie Wen, for his countless hours of reflecting, reading, encouraging, and most of all patience throughout the entire process. Without his support, I could not have done what I was able to do. He was very generous in sharing his experiences on academic research, academic life and beyond. He is not only my adviser, but also, a friend inspiring me for the rest of my life.

Next, I wish to thank my committee members who were more than generous with their expertise and precious time, Dr. Nezih Pala, Dr. Gang Quan, Dr. Jean H. Andrian and Dr. Deng Pan for their support and suggestions in improving the quality of this dissertation. It is truly honored to have such great fantastic and knowledgeable professors serving as my committee members. A special thanks to Dr. Nezih Pala for being friendly, caring and supportive in numerous ways.

I would like to thank all my lab-mates, Tao Liu, Qi Liu and Nuo Xu for all their helps on this research. And I would like to thank the beginning teachers, mentor-teachers and administrators in Department of Electrical and Computer Engineering at Florida International University that assisted me with my graduation.

Finally, I want to thank my family for their unconditional love, faith, and encouragement.

ABSTRACT OF THE DISSERTATION

MACHINE VISION, NOT HUMAN VISION, GUIDED COMPRESSION TOWARDS LOW-LATENCY AND ROBUST DEEP LEARNING SYSTEMS

by

Zihao Liu

Florida International University, 2020

Miami, Florida

Professor Wujie Wen, Major Professor

Deep Neural Networks (DNNs) have been achieving extraordinary performance across many exciting real-world applications, including image classification [28, 57, 106], speech recognition [45, 40], natural language processing [6], medical diagnosis, self-driving cars [12], drones [37], anomaly detection [17] and recognition of voice commands [45]. However, the *de facto* DNN technique in real life exposes to two critical issues:

First, the ever-increasing amounts of data generated from mobile devices, sensors, and the Internet of Things (IoT) challenge the performance of DNN system. there lack efficient solutions to reduce the power-hungry data offloading and storage on terminal devices like edge sensors, especially in face of the stringent constraints on communication bandwidth, energy, and hardware resources.

Second, DNN models are inherently vulnerable to adversarial examples (AEs) [38], i.e. malicious inputs crafted by adding small and human-imperceptible perturbations to normal inputs, strongly fooling the cognitive function of DNNs. Though image compression technique have been explored to mitigate the adversarial examples [35, 29], however, existing solutions are unable to offer a good balance between the efficiency of removing adversarial perturbation on malicious inputs and classification accuracy on benign samples.

This dissertation makes solid strides towards developing low-latency and robust deep learning systems by for the first time leveraging the deep understandings on the image

perception difference between human vision and deep learning systems (a.k.a. "machine vision" in this dissertation). In the first part, we propose to develop three types of “machine vision" guided image compression frameworks, dedicated to accelerating both cloud-based deep learning image classification and 3D medical image segmentation with almost zero accuracy drop, by embracing the nature of deep cascaded information process mechanism of DNN architecture. To the best of our knowledge, this is the first effort to systematically re-architecture existing data compression techniques which are centered around human vision to be machine vision favorable, thereby achieving significant service speed-up. In the second part, we propose a JPEG-based defensive compression framework, namely “feature-distillation”, to effectively rectify adversarial examples without impacting classification accuracy on benign images. Experimental results show that the very low cost “feature-distillation" can deliver the best defense efficiency with negligible accuracy reduction among existing input-preprocessing based defense techniques, serving as a new baseline and reference design for future defense methods development.

TABLE OF CONTENTS

CHAPTER PAGE

1. INTRODUCTION 1

2. DEEPN-JPEG: A DEEP NEURAL NETWORK FAVORABLE JPEG-BASED IMAGE COMPRESSION FRAMEWORK 6
2.1 Background and Motivation 7
2.1.1 Basics of Deep Neural Networks 7
2.1.2 HVS-based JPEG Compression 8
2.1.3 Inefficient HVS Compression for DNNs 10
2.2 Our Approach 11
2.2.1 Modeling the difference of HVS and DNN 11
2.2.2 DNN-Oriented DeepN-JPEG Framework 14
2.2.3 Design Optimization 17
2.3 Evaluation 19
2.3.1 Compression Rate and Accuracy 19
2.3.2 Power Consumption 21
2.4 Conclusion 22

3. MACHINE VISION GUIDED 3D MEDICAL IMAGE COMPRESSION FOR EFFICIENT TRANSMISSION AND ACCURATE SEGMENTATION IN THE CLOUDS 23
3.1 Related Work 24
3.1.1 Fully Convolutional Networks for 3D Segmentation 24
3.1.2 3D Medical Image Compression 25
3.1.3 JPEG-2000 3D image Compression 26
3.2 Machine Vision oriented 3D Image Compression 27
3.2.1 Frequency Analysis Module 28
3.2.2 Mapping Module 31
3.3 Evaluation 33
3.3.1 Experiment Setup 33
3.3.2 Optimal Parameter Selection 34
3.3.3 Comparison of Segmentation Accuracy 35
3.3.4 Comparison of Compression Rate 37
3.3.5 Visual results 38
3.3.6 Overhead 39
3.4 Conclusion 39

4. ORCHESTRATING MEDICAL IMAGE COMPRESSION AND SEGMENTATION NETWORKS FOR EFFICIENT TRANSMISSION AND ACCURATE SEGMENTATION IN THE CLOUDS 42
4.1 Background and Related Work 43

4.1.1 Medical Image Segmentation . . . 43
4.1.2 Medical Image Compression . . . 44
4.2 Our Methodology . . . 46
4.2.1 Architecture Design . . . 47
4.2.2 Training Loss Design . . . 49
4.2.3 Training and Testing . . . 50
4.3 Evaluation . . . 51
4.3.1 Experiment Setup . . . 51
4.3.2 Segmentation Performance . . . 54
4.3.3 Compression Efficiency . . . 57
4.3.4 Cloud-based Service Latency . . . 58
4.3.5 Visual Analysis . . . 60
4.4 Conclusion . . . 60

5. FEATURE DISTILLATION: DNN-ORIENTED JPEG COMPRESSION AGAINST ADVERSARIAL EXAMPLES . . . 62
5.1 Background, Related Work and Motivation . . . 64
5.1.1 Basics of Adversarial Examples and JPEG . . . 64
5.1.2 Related Works . . . 65
5.1.3 Why standard JPEG is not good? . . . 67
5.2 Our Approach–Feature Distillation . . . 67
5.2.1 Step 1: Defensive Quantization for Enhancing Defense . . . 68
5.2.2 Step 2: DNN-Oriented Quantization for Compensating Accuracy Reduction 70
5.3 Evaluation . . . 73
5.3.1 Experimental Setup . . . 74
5.3.2 Optimized Quantization Step . . . 75
5.3.3 Enhanced Robustness Against AE . . . 76
5.4 Conclusion . . . 81

6. CONCLUSIONS . . . 84

BIBLIOGRAPHY . . . 86

VITA . . . 99

LIST OF FIGURES

FIGURE PAGE

2.1 A typical Deep Convolutional Neural Network (DCNN). 8

2.2 Briefly overview of JPEG compression technology 9

2.3 (a) Accuracy v.s. JPEG CRs of “AlexNet” for CASE 1/2; (b) CASE 2– Accuracy w.r.t Epoch Number at various CRs. 10

2.4 Feature degradation will impact the classification. 12

2.5 An overview of heuristic design flow of “DeepN-JPEG" framework. 13

2.6 Parameter optimization for different frequency bands. 17

2.7 Optimization of k_3 parameter in PLM. 18

2.8 The compress rate and accuracy for different methods. 20

2.9 The compress rate and accuracy for different DNN models. 21

2.10 Evaluation of power consumption for different methods. 22

3.1 3D u-Net [33]: a widely used framework in fully convolutional networks for medical image segmentation. 25

3.2 Flow of 3D JPEG-2000 compression method. 26

3.3 Overview of the proposed DNN-oriented 3D image compression framework. 28

3.4 Diverse frequency domain of medical images. 30

3.5 Optimal parameter selection of Q_{max} and Q_{min}. 34

3.6 Compression rate comparison of our method v.s. JPEG-2000 under the same segmentation accuracy. 38

3.7 Segmentation details of four slices in a CT image in HVSRMR 2016 Challenge dataset [78], compressed using our method and JPEG-2000, and segmented by DenseVoxNet [124]. Many details are missing in the segmentation results from JPEG-2000 compressed images but not in our method. Quantitative comparisons can be found in Section 3.3. 41

4.1 Our orchestrating medical image compression and segmentation networks design flow. 44

4.2 Illustration of (a) auto-encoder; (b) discriminator in our design. 47

4.3 (a) Segmentation results under various combinations. (b) Segmentation accuracy/bpp comparison with prior methods. Same bpp (bars): Our (Auto+Seg+Dis); Same dice (line): Our (Auto+Seg+Dis+CR). 56

4.4 Comparison between original and reconstructed (decompressed) images from auto-encoder of 2D RGB images (left 2 columns) and 3D cardiovascular magnetic resonance (CMR) images (right 3 columns) with corresponding predict label and ground truth label. 57

5.1 Illustration of two different modes of “feature distillation"–one pass and two pass. 65

5.2 An overview of heuristic design flow of DNN-Oriented compression based on crafted quantization. 68

5.3 Exploration of the defensive quantization step for a 8*8 table: (a) Defense efficiency of one pass method against adversarial examples; (b) Defense efficiency of two pass method against adversarial examples; (c) Average defense efficiency w.r.t. the legitimate image accuracy (FGSM, $\epsilon = 0.008$); (d) Accuracy impacts of ranked frequency components (FGSM, $\epsilon = 0.008$); (e) Accuracy impacts of various quantization steps w.r.t. different perturbation strength (FGSM). 73

5.4 Defense efficiency of black-box setting for different attack and defense mechanisms on ImageNet. 79

5.5 Visual results produced by default JPEG compression. 82

5.6 Visual results produced by “feature distillation" method. 83

CHAPTER 1

INTRODUCTION

As one of the most fascinating technique when we are entering the era of Artificial Intelligent (AI), Deep Neural Networks (DNNs) are penetrating the real world in many exciting applications such as image classification [28, 57, 106], speech recognition [45, 40], natural language processing [6], medical diagnosis, self-driving cars [12], drones [37], anomaly detection [17] and recognition of voice commands [45] etc. However, when facing real applications, it is always challenging to deliver the low-latency and robust deep learning services.

First, the marriage of *big data* and *deep learning* leads to the great success of artificial intelligence, but it also raises new challenges in data communication, storage, and computation [65, 96] incurred by the growing amount of distributed data and the increasing DNN model size. For resource-constrained IoT applications, while recent researches have been conducted [63, 64, 42] to handle the computation and memory-intensive DNN workloads in an energy efficient manner, *there lack efficient solutions to reduce the power-hungry data offloading and storage on terminal devices like edge sensors, especially in face of the stringent constraints on communication bandwidth, energy, and hardware resources*. Recent studies show that the latencies to upload a JPEG-compressed input image (i.e. 152KB) for a single inference of a popular CNN–"AlexNet" via stable wireless connections with 3G (870ms), LTE (180ms) and Wi-Fi (95ms), can exceed that of DNN computation (6$\sim$82ms) by a mobile or cloud-GPU [55]. Moreover, the communication energy is comparable with the associated DNN computation energy.

Data compression is an indispensable technique that can greatly reduce the data volume needed to be stored and transferred, thus to substantially alleviate the data offloading and local storage cost in terminal devices. As DNNs are contingent upon tons of real-time produced data, it is crucial to compress the overwhelming data effectively. Existing image

compression frameworks (such as JPEG) can compress data aggressively, but they are often optimized for the Human-Visual System (HVS) or human's perceived image quality, which can lead to unacceptable DNN accuracy degradation at higher compression ratios (CR) and thus significantly harm the quality of intelligent services. As shown later, testing a well-trained *AlexNet* using $CR =\sim 5\times$ compressed JPEG images (w.r.t. $CR = 1\times$ high quality images), can lead to $\sim 9\%$ image recognition accuracy reduction for the large-scale dataset— *ImageNet*, almost offsetting the improvement brought by more complex DNN topology, i.e. from *AlexNet* to *GoogLeNet* (8 layers, 724M MACs v.s. 22 layers, 1.43G MACs) [57, 98]. This prompts the need for developing a DNN-favorable deep compression framework.

Moreover, Deep learning has significantly pushed forward the frontier of automatic medical image analysis [24, 84, 118, 20, 24, 19]. On the other hand, most deep learning based frameworks have high computation complexities [113, 119, 117, 120, 116, 53, 54]. For example, the number of operations needed by the network by [22] to segment a 3D Computed Tomography (CT) volume would be around 2.2 Tera (10^{12}) , which needs days to be processed on a general desktop computer. In addition, with the advances in medical imaging technologies, the related data has been increasing exponentially for decades [32]. Ponemon Institute survey found that 30% of the world's data storage resides in the healthcare industry by 2012 [36]. For both reasons, clouds have become a popular platform for efficient deep learning based medical image analysis [71, 126, 114, 115, 1].

Utilizing clouds, however, requires medical images to be transmitted from local to servers. Compared with computation time needed to process these images in the clouds, the transmission time is usually higher. For example, the latency to transmit a 3D CT image of size 300MB is about 13 seconds via fixed broadband internet (estimated with 2017 U.S. average fixed broadband upload speed of 22.79 Mbps [75]). On the other hand, it takes no more than 100 milliseconds for 3D-DSN [33] to segment an image through a

high-performance cluster of 10 GPUs in cloud [70, 27, 55]. For slower internet speed, this gap is even bigger.

To tackle this issue, image compression is typically used to prune unimportant information before sending the image to clouds, thus reducing data traffic. The compression time is usually negligible (e.g., 24 milliseconds to compress a 300MB 3D CT image to 30MB using a moderate GPU [72]). There exist many general image compression standards such as JPEG-2000 [14, 13], JPEG [109], and MPEG2 [51]. Most of these standards use frequency transformation to filter out information that leads to little visual distortion. In addition to the existing 3D image compression standards, alternative compression methods have been proposed in the literature, most of which modify the existing standards to improve their performance [15, 87, 86, 121]. There are also a few methods for lossless compression of 3D medical images [88, 69].

Almost all the existing compression techniques are optimized for the Human-Visual System (HVS), or image quality perceived by humans. However, when we compress images for transmission to the clouds, their quality will not be judged by human vision, but rather by the performance of the neural networks that process them in the clouds. As such, an interesting question that naturally arises is: are the existing compression techniques still optimal for these neural networks. Medical image segmentation extracts different tissues, organs, pathologies, and biological structures to support medical diagnosis, surgical planning and treatments. A critical point in this dissertation is that **deep learning system perceives input images in a completely different way from human vision, given its primary goal is to achieve the best segmentation accuracy via judging the quality of compressed (later decompressed) images by neural networks, rather than the visual distortion of human vision**. As a result, it naturally brings up several interesting questions: *1) Can we design a compression framework optimal for deep learning-based image segmentation instead of human vision? 2) If so, how should we design that? Is it possible*

*to design a **matched** pair of compression and segmentation network guided by the concept of "machine vision" for the whole process? Will the achievable compression ratio and segmentation quality under such a framework outperform existing solutions significantly?*

Second, recent studies have shown that DNN models are inherently vulnerable to adversarial examples (AEs) [39, 99], i.e. malicious inputs crafted by adding small and human-imperceptible perturbations to normal inputs, strongly fooling the cognitive function of DNNs. For example, in image recognition, adversarially manipulating the perceptual systems of autonomous vehicles by physically captured adversarial images, i.e. via camera or sensor [77, 94], can lead to the misreading on road signs, thus causing potential disastrous consequences in DNN-based cyber-physical systems.

Many countermeasures [62, 61, 85, 95, 107, 5] have been proposed to enhance the robustness of DNNs against adversarial examples, mainly including DNN model-specific hardening strategies and model-agnostic defenses [41]. Typical model-specific solutions like "adversarial training" or "defensive distillation" may rectify the model parameters to mitigate the attacks by using the iterative retraining procedure or masking adversarial gradient. The model-agnostic approaches such as input dimensionality reduction [10, 108] or direct JPEG compression [35, 29, 41] attempt to remove adversarial perturbations from the inputs, before feeding them into DNN classifiers.

In this dissertation, to handle the first issue, we for the first time develop a highly efficient image compression framework specifically targeting DNN, on two types of codec engines, i.e. JPEG and JPEG-2000. Moreover, a DNN-based compression neural network, i.e. auto-encoder, has also been explored to further enhance the compression efficiency for compress both 2D and 3D medical images for the machine learning systems without segmentation quality degradation. Unlike existing compressions that are developed by taking the human-perceived distortions as the top priority, our codec based method can preserve important features crucial for DNN classification and segmentation with

guaranteed accuracy and compression rate, thus to drastically lower the cost incurred by data transmission and storage in resource-limited edge devices. For the second issue claimed before, we focus on improving the effectiveness of JPEG compression based model-agnostic defense against adversarial examples in image classification. As we shall show later, directly deploying standard lossy JPEG compression algorithm as a defense method [35, 29] neither effectively removes the adversarial perturbations nor guarantees the accuracy of benign samples. Hence, we for the first redesign the JPEG compression framework to be DNN-favorable (instead of centering around human-visual system (HVS)), and develop a novel low-cost strategy, called *"feature distillation"*, augmented from standard JPEG, to simultaneously improve the DNN robustness against AE attacks while ensuring DNN model's testing accuracy.

The rest of this dissertation is organized as follows. In Chapter 2, we develop an image compression framework tailored for DNN applications, named "DeepN-JPEG", to embrace the nature of deep cascaded information process mechanism of DNN architecture. In Chapter 3, we propose a machine vision guided medical image compression framework for segmentation in the clouds. In Chapter 4, we propose a generative segmentation architecture consisting of a compressive auto-encoder, a segmentation network and a discriminator network. In Chapter 5, we further propose a JPEG-based defensive compression framework, namely"feature distillation", to effectively rectify adversarial examples without impacting classification accuracy on benign data. Finally, in Chapter 6, we conclude this dissertation.

CHAPTER 2

DEEPN-JPEG: A DEEP NEURAL NETWORK FAVORABLE JPEG-BASED IMAGE COMPRESSION FRAMEWORK

As one of most fascinating machine learning techniques, deep neural network (DNN) has demonstrated excellent performance in various intelligent tasks such as image classification. DNN achieves such performance, to a large extent, by performing expensive training over huge volumes of training data. To reduce the data storage and transfer overhead in smart resource-limited Internet-of-Thing (IoT) systems, effective data compression is a “must-have" feature before transferring real-time produced dataset for training or classification. While there have been many well-known image compression approaches (such as JPEG), we for the first time find that a human-visual based image compression approach such as JPEG compression is not an optimized solution for DNN systems, especially with high compression ratios. To this end, we develop an image compression framework tailored for DNN applications, named “DeepN-JPEG", to embrace the nature of deep cascaded information process mechanism of DNN architecture. Extensive experiments, based on “ImageNet" dataset with various state-of-the-art DNNs, show that “DeepN-JPEG" can achieve $\sim 3.5\times$ higher compression rate over the popular JPEG solution while maintaining the same accuracy level for image recognition, demonstrating its great potential of storage and power efficiency in DNN-based smart IoT system design.

In this work, we for the first time develop a highly efficient image compression framework specifically targeting DNN, named DeepN-JPEG. Unlike existing compressions that are developed by taking the human-perceived distortions as the top priority, DeepN-JPEG preserves important features crucial for DNN classification with guaranteed accuracy and compression rate, thus to drastically lower the cost incurred by data transmission and storage in resource-limited edge devices.

Our major contributions are:

1. We propose a semi-analytical model to capture the image processing mechanism differences between the human visual system (HVS) and deep neural network at frequency domain;
2. We develop a DNN-favorable feature refinement methodology by leveraging the statistical frequency component analysis of various image classes;
3. We propose piece-wise linear mapping function to link statistical information of refined features to individual quantization values in the quantization table, thus to optimize the compression rate with minimized accuracy drop.

Experimental results show that DeepN-JPEG can achieve much higher compression efficiency (i.e.$\sim 3.5\times$) than that of JPEG solution while maintaining the same accuracy level with the same hardware cost, demonstrating the great potentials for its applications in low-cost and ultra-low power terminal devices, i.e. edge sensors.

2.1 Background and Motivation

2.1.1 Basics of Deep Neural Networks

DCNN introduces multiple layers with complex structures to model a high-level abstraction of the data [46], as shown in Fig. 2.1 and exhibits high effectiveness in finding hierarchical patterns in high-dimensional data by leveraging the deep cascaded layer structure [44, 57, 92, 98]. Specifically, the convolutional layer extracts sufficient feature maps from the inputs by applying kernel-based convolutions, the pooling layer performs a downsampling operation (through max or mean pooling) along the spatial dimensions for a volume reduction, and the fully-connected layer further computes the class score based on the weighted results and non-linear activation functions. Softmax regression (or multinomial

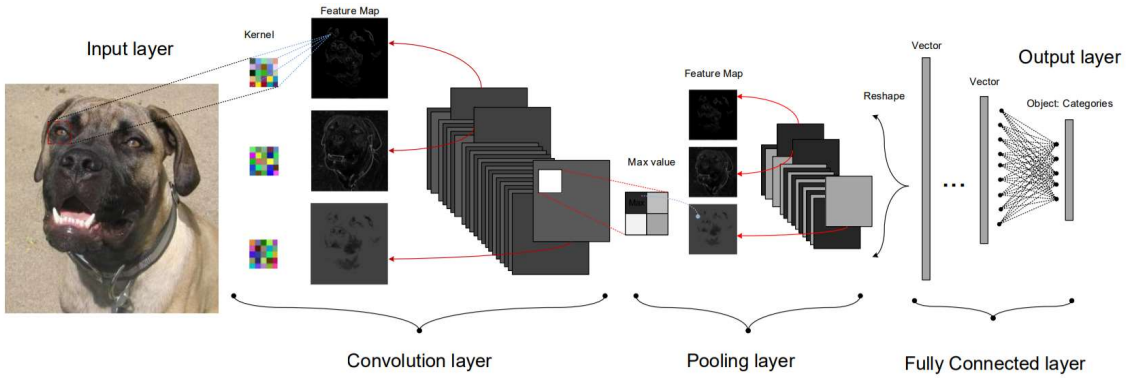

Figure 2.1: A typical Deep Convolutional Neural Network (DCNN).

logistic regression) [11] is usually adopted in the last layer of most DNNs for a final decision.

To perform realistic image recognition, the DNN hyper-parameters are trained extensively through an overwhelming amount of input data. For instance, the large-scale dataset–ImageNet [31], which consists of 1.3 Million high-resolution image samples (~ 140 Gigabyte) in 1K categories, is dedicated to training state-of-the-art DNN models for image recognition task.

2.1.2 HVS-based JPEG Compression

It is widely agreed that massive images and videos, as the major context to be understood by deep neural networks, dominate the wireless bandwidth and storage ranging from edge devices to servers. Hence, in this work, we focus on the image compression.

JPEG [109] is one of the most popular lossy compression standards for digital images. It also forms the foundation of most commonly used video compression formats like Motion JPEG (MPEG) and H.264 etc [82]. As shown in Fig. 2.2, for each color component, i.e. the RGB channels, the input image is first divided into 8×8 non-overlapping pixel blocks, then 2D Fourier Discrete Cosine (DCT) Transform is applied at each 8×8 block

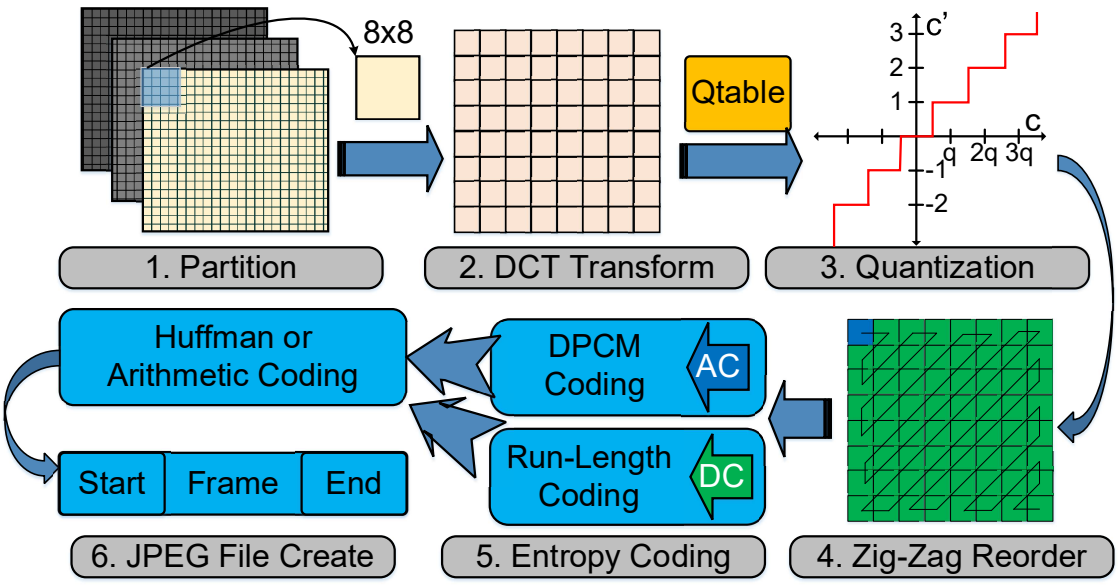

Figure 2.2: Briefly overview of JPEG compression technology .

to generate 64 DCT coefficients $c_{i,j}$, $i \in 0, ..., 7$, $j \in 0, ..., 7$, of which $c_{0,0}$ is direct current (DC) coefficient, and $c_{0,1}, ..., c_{7,7}$ are 63 alternating current (AC) coefficients. Each 64 DCT coefficients are quantized and rounded to the nearest integers as here $q_{i,j}$ is the individual parameter of the 64-element quantization table provided by JPEG [109]. The table is designed to *preserve the low-frequency components and discard high-frequency details because the human visual system (HVS) is less sensitive to the information loss in high-frequency bands* [125]. As a many-to-one mapping, such quantization is fundamentally lossy (i.e. $c_{i,j} \neq c'_{i,j} \times q_{i,j}$ at the decompress stage), and can generate more shared quantized coefficients (i.e. zeros) for a better compression. After quantization, all the quantized coefficients are ordered into the "zig-zag" sequence following the frequency increasing. Finally, the differential coded DC and run-length coded AC coefficients will be further compressed by lossless Huffman or Arithmetic Coding. Increasing (reducing) the compression ratio (CR) can be usually realized by scaling down (up) the quantization table by adjusting the quantization factor (QF). A larger QF indicates better image quality

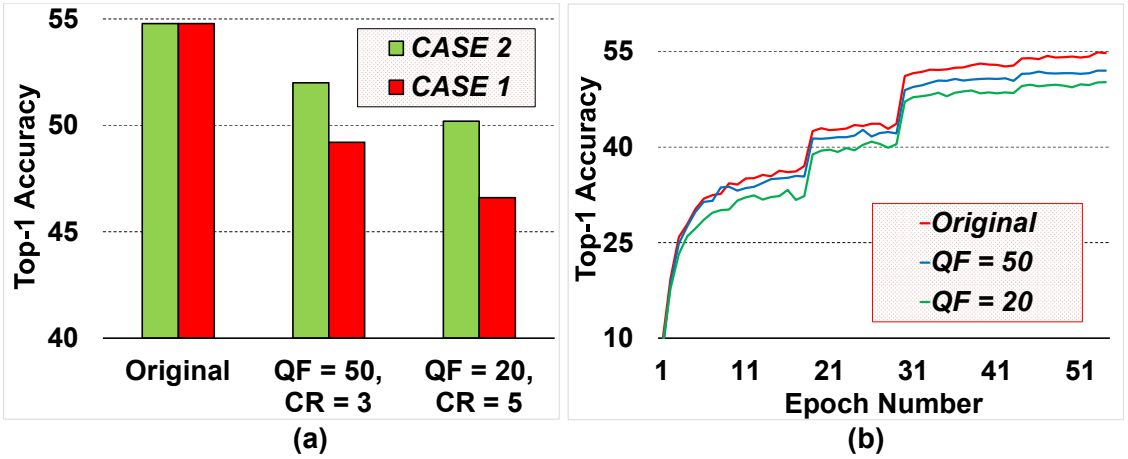

Figure 2.3: (a) Accuracy v.s. JPEG CRs of "AlexNet" for CASE 1/2; (b) CASE 2–Accuracy w.r.t Epoch Number at various CRs.

but a lower CR. A reserved procedure of aforementioned steps can decompress an image.

2.1.3 Inefficient HVS Compression for DNNs

DNN suffers from dramatic accuracy loss if using existing HVS-based compression techniques to aggressively compress the input images for more efficient data offloading and storage: To explore how existing compressions can impact the accuracy of DNN, we have conducted following two sets of experiments: **CASE 1**: training DNN model by high quality JPEG images (QF=100), but testing it with images at various CRs or QFs (i.e. QF=100, 50, 20); **CASE 2**: training DNN model by various compressed images (QF=100, 50, 20), but testing it only with high quality original images (QF=100). In both cases, a representative DNN example–"AlexNet" [57] with 5 convolutional layers, 3 fully connected layers, and 60M weight parameters is trained with the ImageNet dataset for large-scale visual recognition.

As Fig. 2.3 (a) shows, the "top-1" testing accuracies characterized from both cases degrade significantly as the CR increases from 1 to 5 (or QF from 100 to 20). To achieve

the best CR (QF=20, CR=5), the accuracy of CASE 1 (CASE 2) can be even dropped by $\sim 9\%$ ($\sim 5\%$) than that of the original one (QF=100, CR=1). Note that the accuracy improvement of ImageNet from “AlexNet" to “GoogLeNet" is mere $\sim 9\%$, despite the significantly increased number of layers (8 v.s. 22) and multiply-and-accumulate (724M v.s. 1.43G). We also observe that “CASE 2" can always exhibit smaller accuracy reduction than “CASE 1" across all CRs ranging from CR=3 to CR=5. This clearly indicates that training the DNN with more compressed JPEG images (compared with testing ones) can slightly alleviate the accuracy dropping, but cannot completely address this issue. As Fig. 2.3 (b) shows, the accuracy gap between a higher CR (or low QF, i.e. QF=20) and the original one (CR=1) for CASE 2 is maximized at the last testing epoch. Apparently, existing compressions like JPEG, which are centered around the human visual system, are not optimized solutions for DNNs, especially at a higher compression ratio.

2.2 Our Approach

Developing efficient compression frameworks has been widely studied in applications like image and video processing, however, all these researches are taking the human-perceived distortions as the top priority, rather than the unique properties of deep neural networks, such as accuracy, deep cascaded data processing, etc. In this section, we first discover the different views of the human visual system and deep neural network in image processing, and then propose the DNN-favorable JPEG-based image compression framework–“DeepN-JPEG".

2.2.1 Modeling the difference of HVS and DNN

We have initialized our studies on an interesting problem: *What are the major differences of image processing between human vision system (HVS) and deep neural network?* This

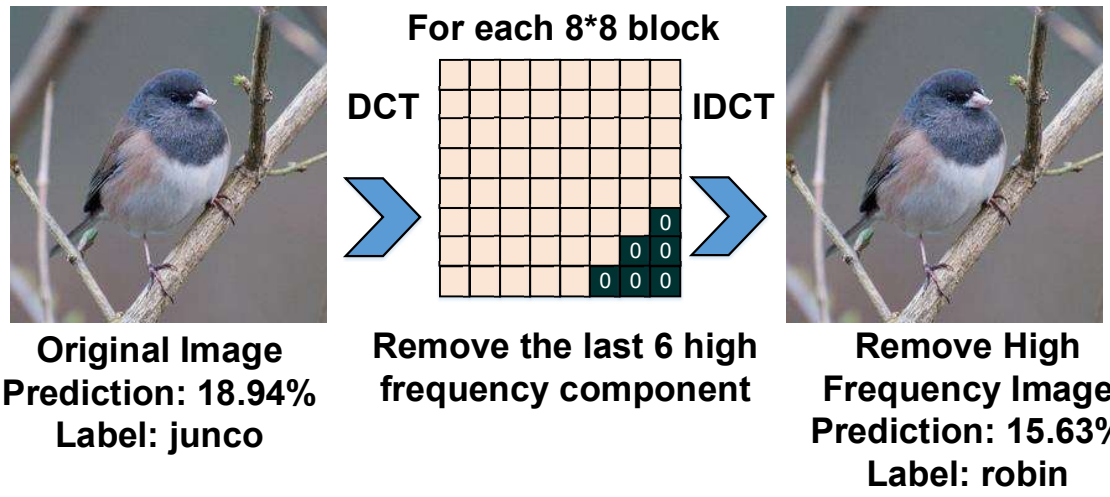

Figure 2.4: Feature degradation will impact the classification.

should help in explaining the aforementioned accuracy reduction issue, thus to guide the development of DNN-favorable compression framework. *Our observation is that DNNs can respond to any important frequency component precisely, but human visual system focuses more on the low-frequency information than high-frequency ones, indicating fewer features to be learned by DNNs after the HVS-inspired compression.* Assume x_k is a single pixel of a raw image $\mathbf{X}$, and x_k can be represented by 8×8 DCT in JPEG compression:

$$x_k = \sum_{i=0}^{i=7} \sum_{j=0}^{n=7} c_{(k,i,j)} \cdot b_{(i,j)} \tag{2.1}$$

where $c_{(k,i,j)}$ and $b_{(i,j)}$ are the DCT coefficient and corresponding basis function at 64 different frequencies, respectively. Because the human visual system is less sensitive to high-frequency components, a higher CR can be achieved in JPEG compression by intentionally discarding the high-frequency parts, i.e. zeroing out the associated DCT coefficient $c_{(k,i,j)}$ through scaled quantization. On the contrary, DNNs examine the importance of the frequency information in a quite different way. The gradient of the DNN function F with respect to a basis function $b_{(i,j)}$ can be calculated as:

$$\frac{\partial F}{\partial b_{(i,j)}} = \frac{\partial F}{\partial x_k} \times \frac{\partial x_k}{\partial b_{i,j}} = \frac{\partial F}{\partial x_k} \times c_{(k,i,j)} \tag{2.2}$$

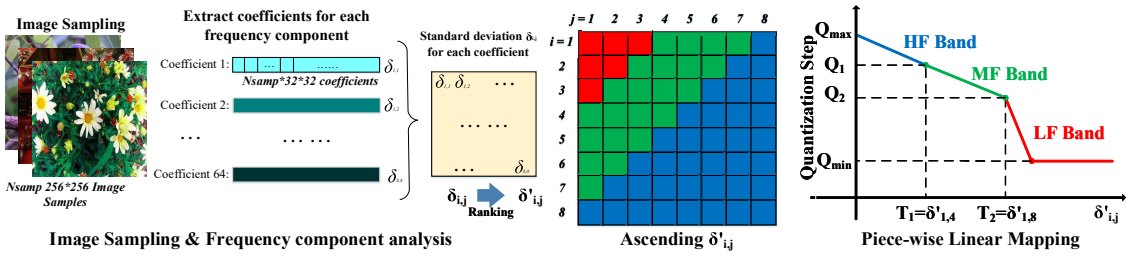

Figure 2.5: An overview of heuristic design flow of "DeepN-JPEG" framework.

Eq. 5.7 implies that the contribution of a frequency component ($b_{i,j}$) of a single pixel x_k to the DNN learning will be mainly determined by its associated DCT coefficient ($c_{(k,i,j)}$) and the importance of the pixel ($\frac{\partial F}{\partial x_k}$). Here $\frac{\partial F}{\partial x_k}$ is obtained after the DNN training, while $c_{(k,i,j)}$ will be distorted by the image compression (i.e. quantization) before training. If $c_{(k,i,j)} = 0$, the frequency feature ($b_{i,j}$), which may carry important details for DNN feature map extraction, cannot be learned by DNN for weights updating, causing a lower accuracy.

It is often the case in a highly compressed JPEG image, given that $c_{(k,i,j)}$s of high-frequency parts (usually small in nature images) are quantized to zero to ensure a better compression rate. As a result, DNNs can easily misclassify aggressively compressed images if their original versions contain important high-frequency features. In CASE 1 (see Fig. 2.3(a)), the DNN model trained with original images learns comprehensive features, especially high-frequency ones that are important in some images, however, such features are actually lost in some more compressed testing images, causing considerable misclassification rate. Fig. 2.4 demonstrates such an example–the "junco" is mis-predicted as "robin" after removing the top six high-frequency components, despite that the differences are almost indistinguishable by human eyes. In CASE 2 (see Fig. 2.3(b)), the model is trained to make decisions solely based on the limited number of features learned from more compressed training images, and the additional features in high quality testing images cannot be detected by DNN for accuracy improvement.

2.2.2 DNN-Oriented DeepN-JPEG Framework

To develop the “DeepN-JEPG" framework, it is essential to minimize the distortion of frequency features that are most important to DNN, thus to maintain the accuracy as much as possible. As quantization is the principal factor to cause important feature loss, i.e. removing less significant high-frequency parts by using a larger quantization step in JPEG, the key step of “DeepN-JEPG" is to re-design such HVS-inspired quantization table to be DNN favorable, i.e. achieving a better compression rate than JPEG without losing needed features.

Although the quantization table redesign has been proved to be a feasible solution in various applications, such as feature detection [18], visual search [34], it is an intractable optimization problem for “DeepN-JPEG" because of the complexity of parameter searching [47], and the difficulty of a quantitative measurement suitable to DNNs. For example, it is non-trivial to characterize the implicit relationship between image feature (or quantization) errors and DNN accuracy loss. Moreover, the characterized results could vary according to the DNN structure. Therefore, it is very challenging to develop a generalized DNN-favorable compression framework.

Our analysis in section 5.2.2 indicates that the contribution of a frequency band to DNN learning is strongly related to the magnitude of the band coefficient. Inspired by this key observation, our “DeepN-JEPG" is developed upon a heuristic design flow (see Fig. 5.2): 1) Sample representative raw images from each class and further characterize the importance of each frequency component through frequency analysis on sampled sub-dataset; 2) Link the statistical information of each feature with the quantization step of quantization table through proposed “Piece-wise Linear Mapping”.

Image Sampling and Frequency Component Analysis

In “DeepN-JPEG" framework, our first step is to sample all classes within the labeled dataset, for more comprehensive feature analysis. To extract the representative features from the whole dataset and rank the importance of those features to DNN, we implied the feature complexity of the image–a smooth image with simple features will be compressed at small size while large size indicates the image consists of more complex features.

As shown in algorithm 1, each sampled image will be first partitioned into $Nblock$ 8×8 blocks, followed by a block-wise DCT. After that, the DCT coefficient distribution at each frequency component will be characterized by sorting all coefficients within the same frequency component across all image blocks collected from different classes of the image dataset. The statistical information, such as the standard deviation $\delta_{i,j}$ of each coefficient, will be calculated based on each individual histogram. Note that such a frequency refinement procedure can precisely tell out the most significant features to DNN, and is different from the simple assumption that low-frequency part is always more important than the high ones can easily lead to the DNN accuracy reduction.

Quantization Table Design

Once the importance of frequency band to DNN is identified by our calibrated DCT coefficient standard deviation, our next question becomes how to link that information to the quantization table design to achieve a higher compression rate with minimized accuracy reduction. The basic idea is to introduce less (more) quantization errors at the critical (less critical) band by leveraging the intrinsic error resilience property of the DNN. To introduce nonuniform quantization errors at different frequency bands, we develop a piece-wise linear mapping function (PLM) to derive the quantization step of each frequency band

Algorithm 1: Frequency component analysis Algorithm

C: # of Classes;
N: # of images in each class;
k: Interval for sampling images;
Spath: Path of Sampled Images;
Nsamp: #number of sampled images;
$fimg_i$: Image in frequency domain;
fc_k: Frequency components;
Nblock: # of 8*8 blocks after block-wise DCT;
δ_k: standard deviation of kth frequency components;
foreach *class* $class_i$ *in [*$class_1$ .. $class_C$*]* **do**
    m = 0; `// count the number of images in certain class`
    foreach *image* img_j *in [*img_1 .. img_N*]* **do**
        m++;
        if *m % k = 0* **then**
            Spath record (Path of img_j)
foreach *image Spath in [*img_1 .. img_{Nsamp}*]* **do**
    $fimg_i$ = 8*8 block-wise DCT (img_i)
    foreach $Block_{i,j}$ *in [1 .. Nblock]* **do**
        $Block_{i,j}$ = $fimg_i$[j*8-8:j*8][j*8-8:j*8]`// ith sampled image jth 8*8 block`
        foreach fc_k *in [1 .. 64]* **do**
            fc_k store $Block_{i,j}$[k]`// ith sampled image jth 8*8 block kth frequency component`
`// Statistical Analysis`
foreach fc_k *in [1 .. 64]* **do**
    calculate standard deviation δ_k
return δ_k `// standard deviation of each frequency components`

from the associated standard deviation:

$$Q_{i,j} = \begin{cases} a - k_1 * \delta_{i,j} & \delta_{i,j} \leq T_1 \\ b - k_2 * \delta_{i,j} & T_1 < \delta_{i,j} \leq T_2 \\ c - k_3 * \delta_{i,j} & \delta_{i,j} > T_2 \end{cases}, \; s.t. \; Q_{i,j} \geq Q_{min} \tag{2.3}$$

where $Q_{i,j}$ is the quantization step at the frequency band (i, j). Q_{min} is the lowest quantization step. a, b, c, k_1, k_2, k_3 are fitting parameters. T_1 and T_2 are thresholds to categorize the 64 frequency bands according to the $\delta'_{i,j}$, i.e. ascending order of the magnitude of $\delta_{i,j}$. As right part of Fig. 5.2 shows, following the similar frequency segmentation in [56], the 64 frequency components are divided into three bands: **Low Frequency (LF)**–1-6 frequency components (largest $\delta'_{i,j}$), **Middle Frequency (MF)**–7-28 and **High Frequency**

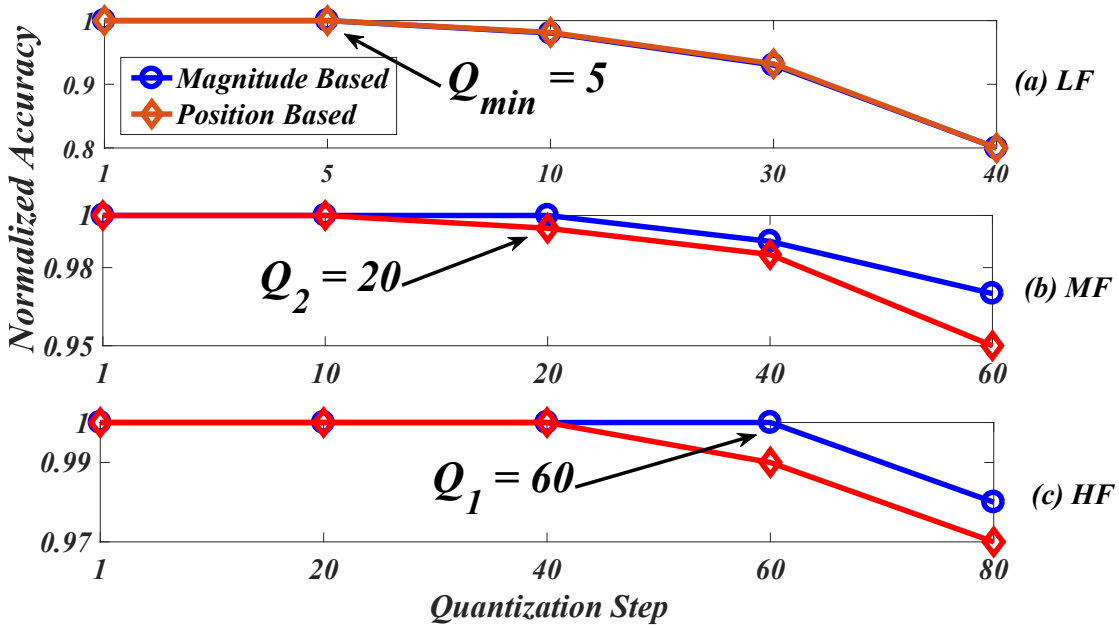

Figure 2.6: Parameter optimization for different frequency bands.

(HF)–29-64 (smallest $\delta'_{i,j}$). Hence, we adopt $T_1 = \delta'_{1,8}$ and $T_2 = \delta'_{1,4}$ in our design. Three different slopes–k_1, k_2, k_3, are assigned to HF band, MF band and LF band, respectively.

2.2.3 Design Optimization

In this section, we explore the parameter optimization for our proposed Piece-wise Linear Mapping based quantization table design. In order to set optimized parameters of Eq. 2.3, i.e. k_1, k_2 and k_3, we first study the sensitivity of quantization steps to DNN accuracy across the LF, MF, and HF bands. We define our proposed band allocation in "DeepN-JPEG" as the "magnitude based", i.e. to segment the frequency band into three types (LF/MF/HF) according to the magnitude of standard deviation of DCT coefficient. For comparison purpose, we also implement the coarse-grained band assignment method based on its position within a default JPEG quantization table, namely "position based". We conduct the simulations by only varying the quantization steps of interested frequency

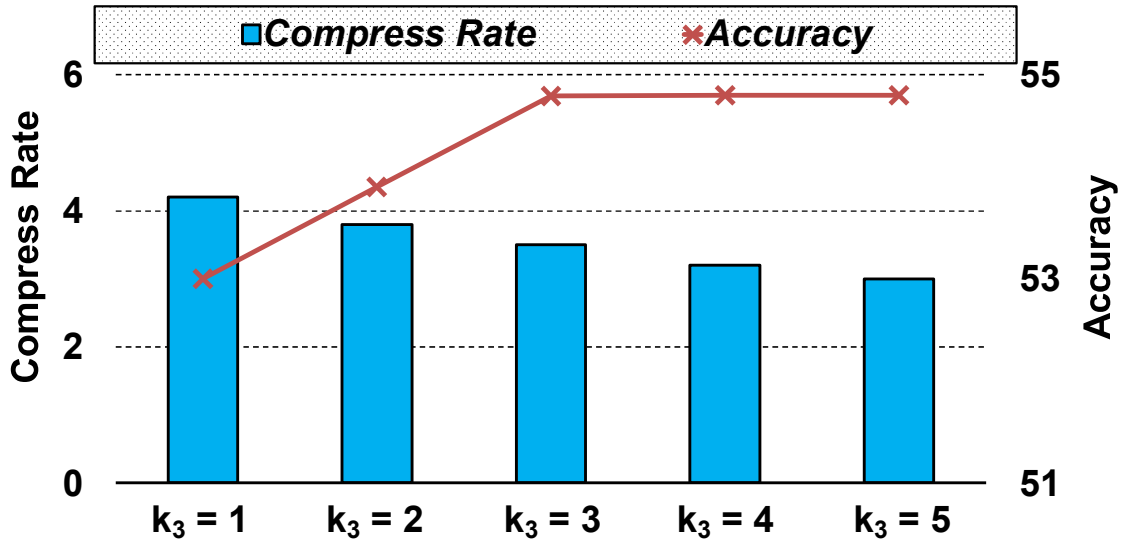

Figure 2.7: Optimization of k_3 parameter in PLM.

bands, while all the others are assigned with minimized quantization steps, i.e. $Q_{i,j} = 1$ without introducing any quantization errors.

Frequency Band Segmentation. As Fig. 2.6 shows, “magnitude based" method can always achieve better accuracy than that of “position based" in both MF and HF bands as the quantization step increases. Moreover, our solution can provide a larger quantization step in both MF and HF bands without accuracy reduction, i.e. 40 v.s. 60 in HF band, which can translate into a higher compression rate than that of JPEG. Besides, we also observe that DNN accuracy starts to drop if $Q_{i,j} > 5$ at the LF band, which indicates that statistically the largest DCT coefficients are most sensitive to quantization errors, thus we set $Q_{min} = 5$ as the lower bound of quantization value to secure the accuracy (see Fig. 2.6 (a)). Similarly, based on the critical points of Fig 2.6 (b) and (c), we can further obtain the quantization steps at the point $T1$ and $T2$, thus to determine the parameters such as k_1, k_2, a and b.

Tuning k_3 in LF Band. Unlike the parameters in MF and HF bands, the optimization of k_3 in LF band is non-trivial because of its significant impact on accuracy and compression

rate. Since k_3 cannot be directly decided according to the lower bound Q_{min} and c, we investigate the correlation between compress rate and accuracy based on a variety of k_3. As shown in Fig. 2.7, a smaller k_3 can offer a better compression rate by slightly sacrificing the DNN accuracy. Based on our observation, we choose $k_3 = 3$ to maximize the compression rate while maintaining the original accuracy.

2.3 Evaluation

Our experiments are conducted on the deep learning open source framework Torch [26]. The “DeepN-JPEG" framework is implemented by heavily modifying the open source JPEG framework [49].

The large-scale ImageNet [31] dataset is adopted to measure the improvement of compression rate and classification accuracy. Specifically, all images are maintained as their original scales in our evaluation without any speed-up trick such as resize or pre-processing. The optimized parameters of “DeepN-JPEG" framework dedicated to ImageNet are as follows: $a = 255,\ b = 80,\ c = 240,\ T_1 = 20,\ T_2 = 60,\ k_1 = 9.75,\ k_2 = 1,\ k_3 = 3$. Four state-of-the-art DNN models are evaluated in our experiment–AlexNet [57], VGG [92], GoogLeNet [98] and ResNet [44].

2.3.1 Compression Rate and Accuracy

We first evaluate the compression rate and classification accuracy of our proposed DeepN-JPEG framework. Three baseline designs are implemented for comparison purpose: the “original" dataset compressed by JPEG (QF=100, CR=1), “RM-HF” compressed dataset and “SAME-Q” compressed dataset. Specifically, “RM-HF” is extended from JPEG by removing the *top-N high-frequency components* from the quantization table to further

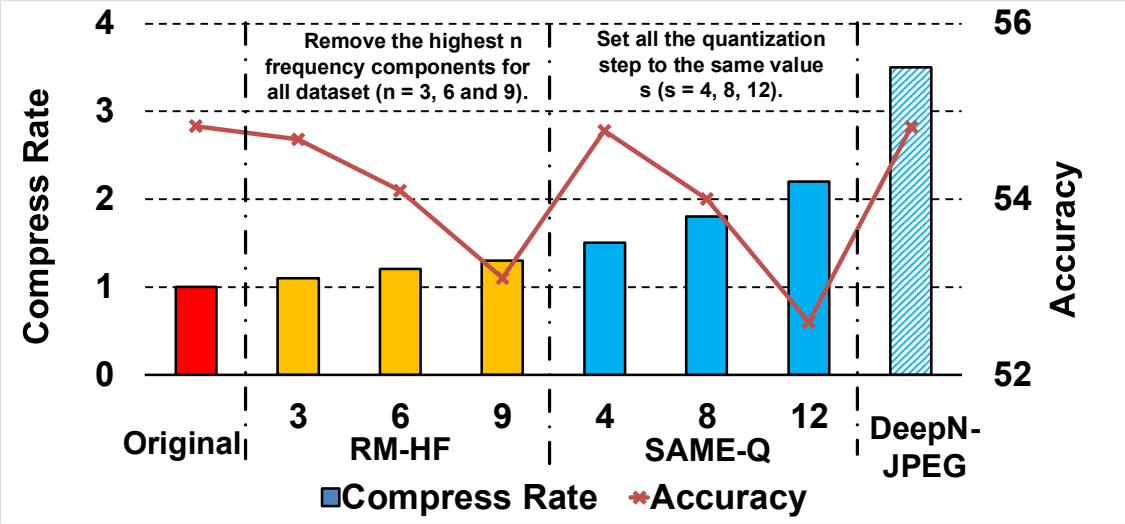

Figure 2.8: The compress rate and accuracy for different methods.

improve the compression rate, and "SAME-Q" denotes a more aggressive compression method with the same quantization step Q for all frequency components.

Fig. 2.8 compares the compression rate and accuracy based on the "ImageNet" dataset "AlexNet" DNN model for all selected candidates. Compared with the "original", "RM-HF" slightly increases the compression rate ($\sim 1.1\times - \sim 1.3\times$) by removing more highest frequency components (top-3–top-9), while "SAME-Q" achieves better compression rates ($\sim 1.5\times - \sim 2\times$). However, both schemes suffer from increased accuracy reduction (w.r.t. "original") as long as the compression rate becomes higher. On the contrary, our "DeepN-JPEG" delivers the best compression rate (i.e. $\sim 3.5\times$) while maintaining the similar high accuracy as that of the original dataset, indicating a promising solution to reduce the cost of data traffic and storage of edge devices for deep learning tasks.

Generality of DeepN-JPEG. We also extend our evaluations across several state-of-the-art DNNs to study how the "DeepN-JPEG" framework responses to different DNN architectures, including GoogLeNet, VGG-16, ResNet-34, and ResNet-50. As shown in Fig. 2.9, our proposed "DeepN-JPEG" can always maintain the original accuracies (w.r.t.

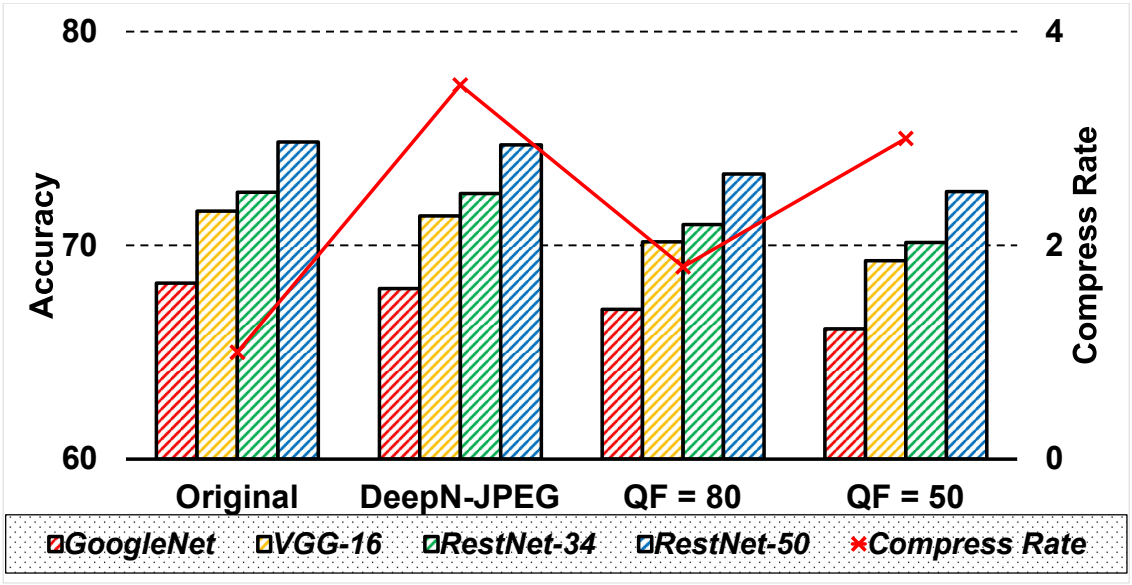

Figure 2.9: The compress rate and accuracy for different DNN models.

"Original") for all selected DNN models. Although JPEG can easily achieve a similar compression rate as that of "DeepN-JPEG" by largely reducing the JPEG QF value, e.g. $QF \leqslant 50$, such an aggressive "data lossy" compression results in significant side effect on the classification performance of all selected DNN models. In contrast, "DeepN-JPEG" can preserve both high compression rate and accuracy for all DNNs, thus a generalized solution.

2.3.2 Power Consumption

In resource-constraint terminal devices, the data offloading incurred power consumption can even outperform that of DNN computation in deep learning [55]. Date compression can reduce the associated cost. Following the same measurement in [55], Fig. 2.10 shows the results of power reduction breakdown. Our "DeepN-JPEG" based data processing consumes only 30% energy without accuracy reduction when compared with that of the original dataset. Compared with "RM-HF3" (remove the top-3 high-frequency components

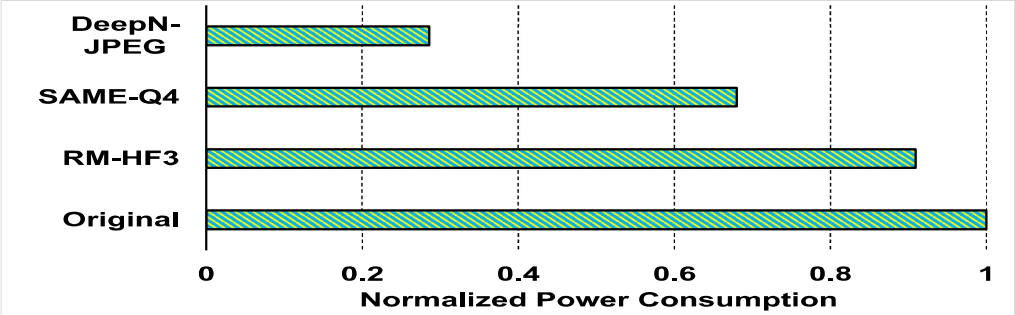

Figure 2.10: Evaluation of power consumption for different methods.

in quantization table) and “SAME-Q4” (the same quantization value–4 in quantization table), “DeepN-JPEG" can still achieve $\sim 2\times$ and $\sim 3\times$ power reduction respectively, due to more efficient data compression.

2.4 Conclusion

The ever-increasing data transfer and storage overhead significantly challenge the energy efficiency and performance of large-scale DNNs. In this chapter, we propose a DNN oriented image compression framework, namely “DeepN-JPEG", to ease the storage and data communication overhead. Instead of the Human Vision System inspired JPEG compression, our solution effectively reduces the quantization error based on the frequency component analysis and rectified quantization table, and further increases the compressing rate without any accuracy degradation. Our experimental results show that “DeepN-JPEG” achieves $\sim 3.5\times$ compression rate improvement, and consumes only 30% power of the conventional JPEG without classification accuracy degradation, thus a promising solution for data storage and communication for deep learning.

CHAPTER 3

MACHINE VISION GUIDED 3D MEDICAL IMAGE COMPRESSION FOR EFFICIENT TRANSMISSION AND ACCURATE SEGMENTATION IN THE CLOUDS

Cloud based medical image analysis has become popular recently due to the high computation complexities of various deep neural network (DNN) based frameworks and the increasingly large volume of medical images that need to be processed. It has been demonstrated that for medical images the transmission from local to clouds is much more expensive than the computation in the clouds itself. Towards this, 3D image compression techniques have been widely applied to reduce the data traffic. However, most of the existing image compression techniques are developed around human vision, i.e., they are designed to minimize distortions that can be perceived by human eyes. In this chapter we will use deep learning based medical image segmentation as a vehicle and demonstrate that interestingly, machine and human view the compression quality differently. Medical images compressed with good quality w.r.t. human vision may result in inferior segmentation accuracy. We then design a machine vision oriented 3D image compression framework tailored for segmentation using DNNs. Our method automatically extracts and retains image features that are most important to the segmentation. Comprehensive experiments on widely adopted segmentation frameworks with HVSMR 2016 challenge dataset show that our method can achieve significantly higher segmentation accuracy at the same compression rate, or much better compression rate under the same segmentation accuracy, when compared with the existing JPEG 2000 method. To the best of the authors' knowledge, this is the first machine vision guided medical image compression framework for segmentation in the clouds.

In this chapter, we propose a machine vision guided 3D image compression framework tailored for deep learning based medical image segmentation in the clouds. Different

from most existing compression methods that take human visual distortion as guide, our method extracts and retains features that are most important to segmentation, so that the segmentation quality can be maintained. We conducted comprehensive experiments on two widely adopted segmentation frameworks (DenseVoxNet [124] and 3D-DSN [33] using the HVSMR 2016 Challenge dataset [78]. Examples on the qualitative effect of our method on the final segmentation results can be viewed in Fig. 3.7.

The main contributions of our work are as follows:

- We discovered that for medical image segmentation in the clouds, traditional compression methods guided by human vision will result in inferior accuracy, and a new method guided by machine vision is warranted.
- We proposed a method that can automatically extract important frequencies for neural network based image segmentation, and map them to quantization steps for better compression.
- Experimental results show our method outperforms JPEG-2000 in two aspects: for a same compression rate, our method achieves significantly improved segmentation accuracy; for a same level of segmentation accuracy, it offers much higher compression rate ($3\times$). These advantages demonstrate great potentials for its application in today's deep neural network assisted medical image segmentation.

3.1 Related Work

3.1.1 Fully Convolutional Networks for 3D Segmentation

Fully convolutional networks (FCN) is a special category of DNN, which is widely used for 3D medical image segmentation. Compared with general DNNs, FCNs only has convolutional layers, up convolutional layer, and pooling layers as shown in Fig. 3.1.

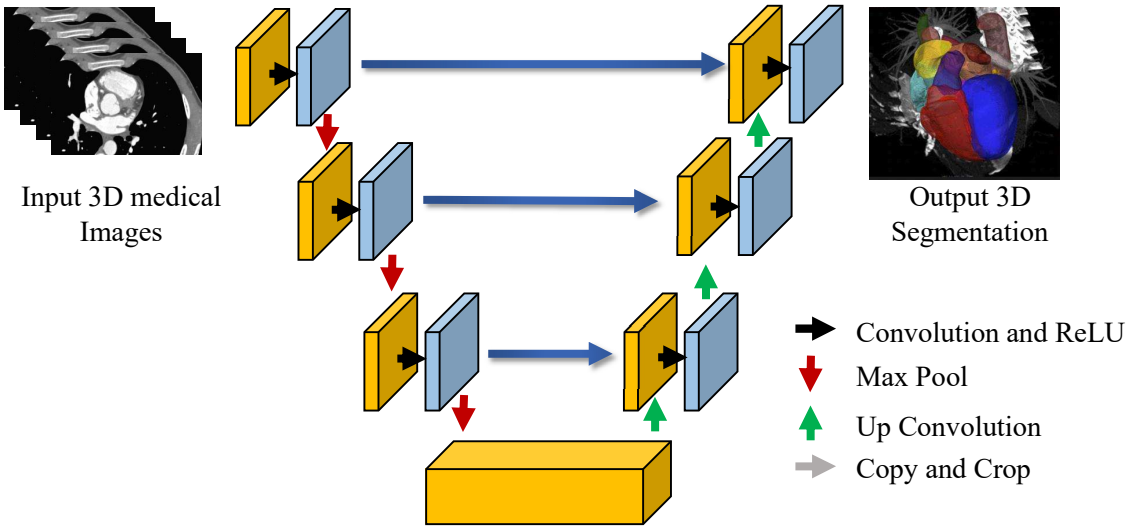

Figure 3.1: 3D u-Net [33]: a widely used framework in fully convolutional networks for medical image segmentation.

With this characteristics, FCNs can efficiently output images with the same size as the input images as shown in Fig. 3.1, which is extremely efficient for segmentation. Almost all the DNN based methods for 3D image segmentation adopt FCN as the backbone network structure, and add some structure and training strategy improvement. For example, 3D U-Net adds more connections between the first several layers and the last several layers as shown in Fig. 3.1 to better extract features. Please refer to related literature [20, 24, 73, 19, 33] for more details of such improvements.

3.1.2 3D Medical Image Compression

There are many general image compression standards such as JPEG-2000 [14][13], JPEG [109]. Some video coding standards such as H.264/AVC [101],and MPEG2 [51] can also be adopted for 3D image segmentation. Most of these standards use transforms such as Discrete Cosine Transform (DCT) and Discrete Wavelet Transform (DWT) for compression while preserving important visual information for humans.

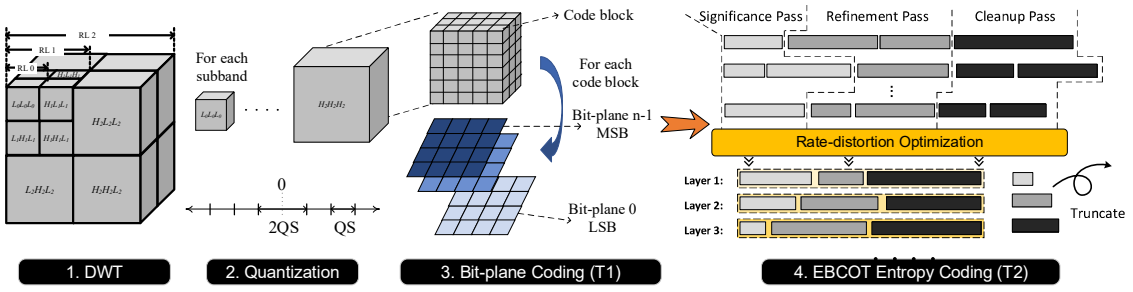

Figure 3.2: Flow of 3D JPEG-2000 compression method.

In addition to the existing 3D medical image compression standard, alternative compression methods have been proposed in the literature. Most of the methods modified the existing standards to improve its performance. Bruylants [15] adopted volumetric wavelets and entropy-coding to improve the compression performance. Sanchez [87] employed a 3-D integer wavelet transform to perform column of interest coding. Sanchez [86] reduced the energy of the sub-bands by exploiting the anatomical symmetries typically present in structural medical images. Zhongwei [121] improved the compression performance by removing unimportant image regions not required for medical diagnosis. There are a few methods for lossless compression of 3D medical images. Santos [88] processed each frame sequentially and using 3D predictors based on the previously encoded frames. Lucas [69] further adopted 3D block classification to process the data at the volume level.

Almost all the above methods still adopt the same objective as that used by JPEG-2000, i.e., to minimize human perceived distortions. As shown in the example in Fig. 3.7, when it comes to the deep learning based segmentation, such a strategy may lead to poor accuracy.

3.1.3 JPEG-2000 3D image Compression

Our method is also based on JPEG-2000 but modifies its human vision guided objective to one that is guided by the segmentation network. Here we briefly review the details of JPEG-2000 so that later we can explain our work better. Fig. 3.2 shows the major

steps in JPEG-2000 compression: First, the 3D discrete wavelet transform (DWT) is applied to an image to decompose it into a multiple-resolution representation in frequency domain [91][7][80]. For example, a 3-D wavelet decomposition leads to three resolution levels (L1, L2, L3). Each resolution level (except L1) is composed of eight subbands: subband 1 to subband 8. The eight lower resolution levels are always generated by progressively applying the 3D DWT process to the upper-left-front block (e.g., subband 1) from the previous resolution level. Then a non-uniform quantization process is applied to each subband based on the number of low pass filters in the subband:

$$x' = \lfloor \frac{x}{QS} \rfloor \tag{3.1}$$

where x is the original coefficient after 3D DWT, QS is the quantization step of a subband and x' is the coefficient after quantization.

The rule is that the more low pass filters a subband has the smaller quantization step are applied to the corresponding subband. This is because Human Visual System (HVS) is more sensitive to low pass frequency information, thus less quantization errors in low pass subband. Bit-plane coding and entropy coding mainly perform coding and please interested readers are referred to the related literature [100][89][93][100] for more details.

3.2 Machine Vision oriented 3D Image Compression

In this section, the details of the proposed machine vision oriented 3D image compression framework for segmentation in the clouds is presented. As shown in Fig. 3.3, the framework contains two modules: frequency analysis module and mapping module. Compared with original JPEG-2000 compression method, the added two modules can obtain optimized quantization steps (QSs) for better segmentation accuracy. The frequency analysis module extracts frequencies important to segmentation with high statistic indexes (SI) using a

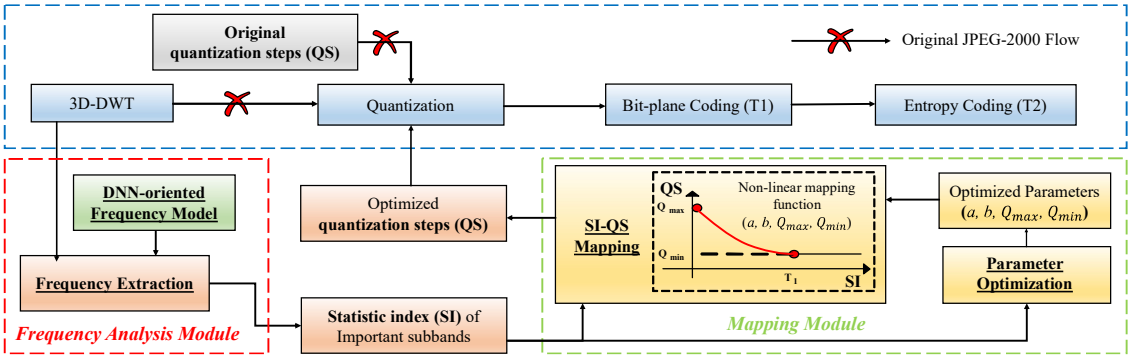

Figure 3.3: Overview of the proposed DNN-oriented 3D image compression framework.

machine vision guided frequency model. The mapping module maps these SIs to optimized QSs which are further provided to the quantization module in JPEG-2000 flow for the rest of the processing. Particularly parameter optimization is also performed to find the optimal parameters in the mapping module.

3.2.1 Frequency Analysis Module

Machine Vision Guided Frequency Model

In this section we build a frequency model that identifies information most useful for segmentation. Assume x_i is a single voxel of a raw 3D image $\mathbf{X}$. x_i can be represented by 3D-DWT at one resolution level in JPEG-2000 compression as:

$$x_i = \sum_{n=0}^{i=N-1} c_i^n \cdot b_i^n \tag{3.2}$$

where c_i^n and b_i^n are the 3D-DWT coefficient at matching 3D coordinate i and corresponding basis function at N different subbands, respectively.

For human visual system, the quantization step (QS) for each subband in JPEG-2000 is positively correlated with the number of high pass filters in a subband. For example, the QS of subband 4 is larger than that of subband 2 at the same resolution level. Then

larger QS in high frequency subband will increase the distortion of coefficients in this subband. Consequently, it will either directly zero out the associated 3D-DWT coefficient c_i^n or increase the chance to truncate them at rate-distortion optimization process. This is because HVS is less sensitive to high frequency subband, so a high compression rate can be achieved by discarding the high frequency information. In order to obtain the important frequency for DNN based segmentation, we calculate the gradient of the DNN loss function F with respect to a basis function b_i^n as:

$$\frac{\partial F}{\partial b_i^n} = \frac{\partial F}{\partial x_i} \times \frac{\partial x_i}{\partial b_i^n} = \frac{\partial F}{\partial x_i} \times c_i^n. \tag{3.3}$$

Equation (5.7) indicates that the importance of information at different subbands of a single voxel to DNN is determined by its associated 3D-DWT coefficients (c_i^n) at all subbands. This is quite different from HVS which distorts c_i^n in high frequency subbands (i.e. quantization or truncation). Large c_i^n in high frequency subband will be heavily distorted in JPEG-2000. However, it may carry important information for DNN segmentation, causing accuracy degradation.

Frequency Extraction

In this section, we extract important frequencies based on the above frequency model. Previous studies [97][60] have demonstrated that the distribution of un-quantized 3D-DWT coefficients in a subband indicates the energy in this subband. Moreover, the distribution of each subband has been proven that they approximately obey a Laplace distribution with zero mean and different standard deviations (δ_n). The larger δ_n a subband has (i.e. more energy in this subband), the more contribution this subband will provide to DNN results. Therefore, δ_n of each subband after 3D-DWT are selected as the SI to represent the importance to DNN. Based on this we propose to conduct the frequency analysis as follows: the number of subbands will be first calculated based on the number of resolution levels

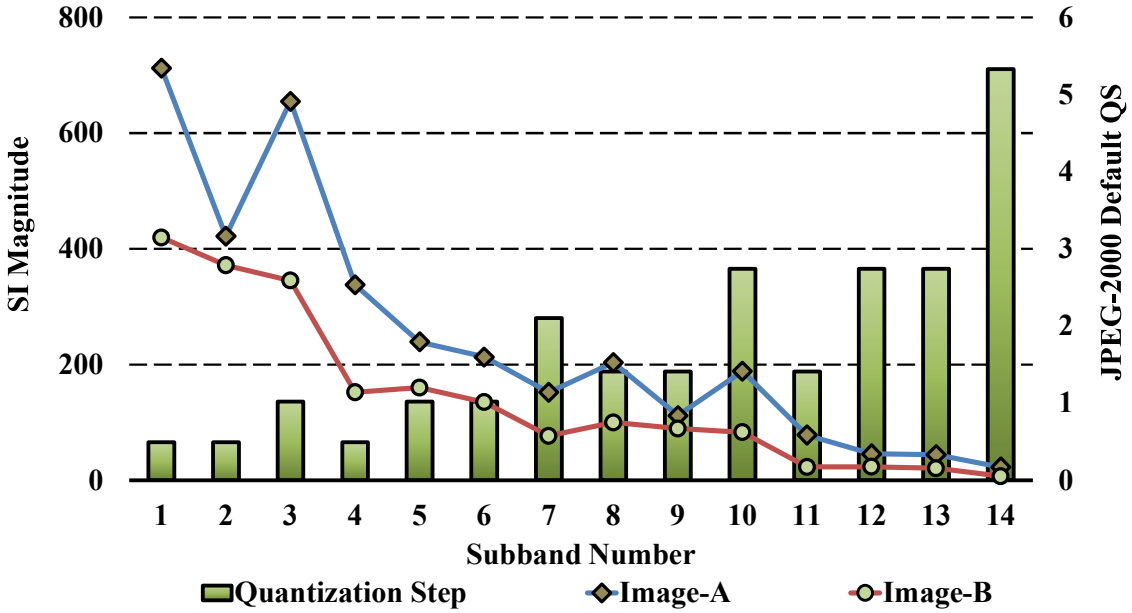

Figure 3.4: Diverse frequency domain of medical images.

at three different dimensions provided by users. After that coefficients that belong to the same subband will be grouped up and reshaped to one dimension. Then the distributions of reshaped coefficients at each subband will be characterized. Finally, the statistical information of each subband, i.e. the standard deviation or SI, will be calculated based on its histogram. The results from this frequency information projection procedure can clearly indicate the importance of each subband to DNN by its SI. With the above discussion, we further analyzed SIs and QSs in JPEG-2000 to show that JPEG-2000 is not optimized for DNNs. We randomly selected two images from HVSMR 2016 dataset labeled as A and B, and then applied our frequency extraction method on them after 3-3-3 3D-DWT. As shown in Fig. 3.4, some important subbands have large QS which is undesired. For example, subband 2 is less important than subband 3 for image A since $\delta_2 < \delta_3$, however, its QS is much smaller than that of subband 3. The same problem exists for subband 3 and subband 4 with image B. Thus, although lower frequency information is always more important than that of higher frequency in JPEG-2000, it is not the case for segmentation accuracy.

Table 3.1: Segmentation results of our methods and JPEG-2000 using DenseVoxNet and HVSMR2016 dataset. The compression rate is set to 30 for both techniques. The images compressed by ours can be segmented with almost the same accuracy as, or sometimes even better than the original ones, much better than those compressed by JPEG-2000. The segmentation performance of NLM is very close to or even better than that with the original images while is much better than JPEG-2000.

		Original	Ours	JPEG-2000
Myocardium	**Dice**	0.838±0.0334	**0.834±0.0386**	0.816±0.042
	Hausdorff	30.879±7.592	**31±7.940**	33.513±7.566
	ASD	0.673±0.67	**0.652±0.671**	0.722±0.746
Blood Pool	**Dice**	0.915±0.025	**0.914±0.024**	0.912±0.025
	Hausdorff	41.034±9.326	**40.93±9.52**	41.031±9.648
	ASD	0.601±0.455	**0.556±0.432**	0.582±0.453
Compression Rate		1	**∼30x**	**∼30x**
PSNR (dB)		∞	∼35	**∼36**

3.2.2 Mapping Module

SI-QS Mapping

With SIs at each subband, our next step is to find a suitable mapping between SI and QS by well leveraging the intrinsic error resilience characteristic of DNN computation. As a result, the segmentation accuracy loss due to increasing compression rate, can be minimized by largely quantilizing the frequency subbands that are less significant to DNN.

In order to precisely model the mapping, we attempt to find a QS curve aligning with most of the SIs. With extensive experiments (we add these experiments in the supplemental material), we observe that the QS-SI points obey a reciprocal function ($y = 1/x$). Thus, we propose a non-linear mapping (NLM) method to implement nonuniform quantization steps at different subbands:

$$Q_n = \frac{a}{(\delta_n + b)}, \quad s.t. \quad Q_{min} \leq Q_n \leq Q_{max} \tag{3.4}$$

where Q_n is the quantization step at subband n, Q_{min} and Q_{max} are the smallest and largest QS, and a and b are the fitting parameters.

Table 3.2: Segmentation results of our methods and JPEG-2000 using 3D-DSN and HVSMR 2016 dataset. The compression rate is set to 30 for both techniques. The images compressed by ours can be segmented with almost the same accuracy as the original ones, and significantly better than those compressed by JPEG-2000.

		Original	Ours	JPEG-2000
Myocardium	**Dice**	0.784±0.059	**0.786±0.059**	0.773±0.058
	Hausdorff	32.345±9.164	**31.002±8.988**	33.041±8.768
	ASD	0.310±0.171	**0.325±0.184**	0.355±0.224
Blood Pool	**Dice**	0.909±0.027	**0.908±0.030**	0.901±0.032
	Hausdorff	38.515±9.59	**38.601±9.951**	39.416±9.932
	ASD	0.235±0.200	**0.223±0.201**	0.230±0.204
Compression Rate		1	**∼30x**	**∼30x**
PSNR (dB)		∞	∼35	**∼36**

Parameter Optimization

With the proposed mapping function, parameter optimization is performed to obtain the optimal a, b, Q_{max} and Q_{min} in Equation (3.4). For a and b, we found that rational functions can fit the relationship between the standard deviation of each subband of an image and the quantization step very well. For Q_{max} and Q_{min}, we examine two corner cases, i.e. upper/lower corner to explore the quantization error tolerance for the most insignificant/significant subband. Then all the parameters in non-linear mapping method can be calculated by substituting pairs (Q_{min}, δ_{max}) and (Q_{max}, δ_{min}) into Equation (3.4).

Lower Corner Case:

we assign the same QS to all the subbands to explore As long as the error induced by QS in the subband with δ_{max} (the most significant subband to DNN) does not impact the segmentation accuracy, this will also hold true for all the other subbands.

Upper Corner Case: To find Q_{max}, we only vary QS at the subband with δ_{min}, while fixing that of all the other subbands as the same QS–Q_{min}. If the subband with δ_{min} (the least significant subband to DNN) cannot tolerate the error incurred by a Q_{max}, the other subbands cannot either.

3.3 Evaluation

3.3.1 Experiment Setup

Our proposed machine vision guided 3D image compression framework was realized by heavily modifying the open-source JPEG-2000 code [2]. This code also served as our baseline–JPEG-2000 for comparison.

Benchmarks: we adopted the HVSMR 2016 Challenge dataset [78] as our evaluation benchmark. This dataset consists of in total 10 3D cardiac MR scans for training and 10 scans for testing. Each image also includes three segmentation labels: myocardium, blood pool, and background.

Evaluation Metrics: We compared our method with the baseline (JPEG-2000) in following two aspects: 1) segmentation results; 2) compression rate. For the segmentation results, we followed the rule of HVSMR 2016 challenge where the results are ranked based on *Dice coefficient (Dice)*. The other two ancillary measurement metrics, i.e. *average surface distance (ASD)* and *symmetric Hausdorff distance (Hausdorff)*, were also calculated for reference. Among the three metrics, a higher Dice represents higher agreement between the segmentation result and the ground truth, while lower ASD and Hausdorff values indicate higher boundary similarity.

Experiment Methods: To evaluate our methods comprehensively, two state-of-art segmentation neural network models–DenseVoxNet [124] and 3D-DSN [33] were selected. We followed the original settings of the two frameworks at training and testing phases but with compressed images. In the testing phase, since the ground truth labels of the selected dataset are not publicly available, we randomly selected five un-compressed training images for training and the rest compressed five for testing. All our experiments were conducted on a workstation which hosts NVIDIA Tesla P100 GPU and deep learning framework Caffe [52] integrated with MATLAB programming interface.

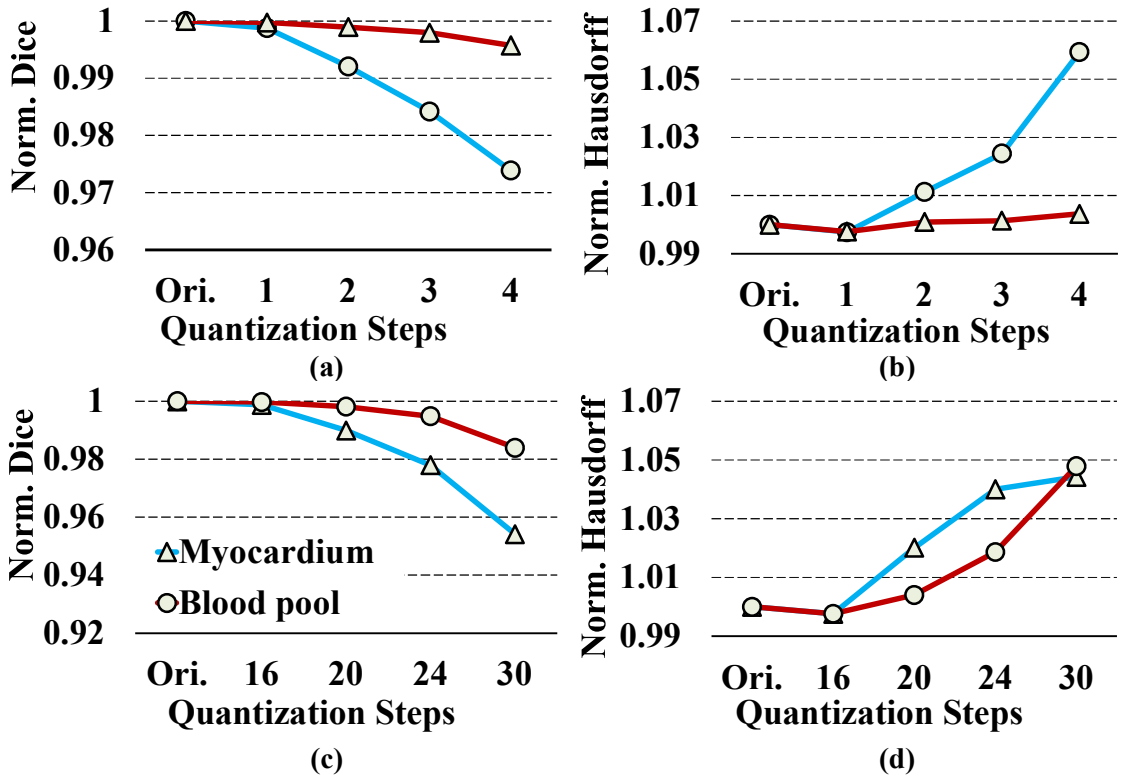

Figure 3.5: Optimal parameter selection of Q_{max} and Q_{min}.

3.3.2 Optimal Parameter Selection

In this section, we experimentally find the optimal parameters for Q_{max}, Q_{min}, a and b in Equation (3.4), following the method discussed in Section 3.2.2. We tested the two cases as discussed in Section 3.2.2 to find Q_{max}, Q_{min}. We took normalized dice coefficients and Hausdorff distance as segmentation measurements for an 3D cardiac MR scan and adopted the FCN model–DenseVoxNet. The measurements for two classes– myocardium and blood pool, are reported. For the lower corner case, as Fig. 3.5 (a) and (b) show, the two measurements for both labels do not suffer from any degradation only if QS is not larger than 1. Therefore, $Q_{min} = 1$ should be selected as ensure the segmentation results. For the upper corner case, the results are shown in Fig. 3.5 (c) and (d). The two measurements decrease when the QS at δ_{min} is larger than 16 at both classes, by

following a similar trend as the lower corner case. Hence, we chose $Q_{max} = 16$ as the upper bound for our quantization step. Based on Q_{max}, Q_{min} and Equation (3.4), a and b can be decided accordingly. In our evaluation, we only adopt DenseVoxNet, as an example, to obtain Q_{max}, Q_{min} so as to solve a and b in Equation (3.4). Then we directly apply it to both DenseVoxNet and 3D-DSN. Note that our method is model agnostic (or rather data specific), since Equation (5.7) indicates that the importance of subbands can largely rely on DWT coefficients without correlating with DNN model. Therefore, we can use the same tuned parameters in our compression regardless of network structure. This is also one of the advantages of our method.

3.3.3 Comparison of Segmentation Accuracy

We first evaluated how our proposed compression framework can improve the segmentation accuracy over the baseline–3D JEPG-2000 using the state-of-the-art segmentation neural network model–DenseVoxNet. For a fair comparison, both our method and 3D JPEG-2000 were implemented at the same compression rate (CR). For illustration purpose, we only report the segmentation accuracy at $CR = 30\times$ (results under other compression rates are summarized in the supplemental material). The mean and standard deviation of the three segmentation measurement metrics–Dice, ASD and Hausdorff, are calibrated from the 5 testing images of HVSMR2016 dataset. Note that Dice is the most important metric among the three.

Table 3.1 reports the segmentation results of the two classes–myrocardium and blood pool for the three methods–original (uncompressed, $CR = 1\times$), ours and JPEG-2000, under DenseVoxNet. **First**, the default 3D JPEG-2000 exhibits the worst segmentation results at all the three metrics among the three methods. This is as expected, since JPEG-2000 takes the human perceived image quality as the top priority by offering the highest

PSNR (~ 36). **Second**, our method, which is developed upon the "machine vision", can beat JPEG-2000 across all three metrics for both classes, with a lower PSNR (~ 35). Impressively, for myocardium, our method can significantly improve Dice, Hausdorff and ASD over JPEG-2000 by 0.018, 2.039, 0.3 on average, respectively. The improvements on blood pool, on the other hand, are relatively limited, given its much higher dice score (0.915 for blood pool v.s. 0.838 for myocardium). **Third**, compared with the original image for both classes, our method only slightly degrades the segmentation results, i.e. $0.001 \sim 0.004$ on average for Dice, but offers a much higher compression rate ($30\times$ v.s. $1\times$). We also observe that the degradation of all three metrics on compressed images of myocardium (w.r.t. original) is always more significant than blood pool, for both our method and JPEG-2000. This is because myoscardium has a lower dice score than blood pool due to the ambiguous border. These results are consistent with the previous work [124].

We would like to emphasize that the achieved performance improvement of our method is very significant for segmentation on the HVSMR 2016 Challenge dataset [110][124] (we also add detailed image by image segmentation results in the supplemental material). Tens of studies performed extensive optimization for segmentation on this dataset. While DenseVoxNet offers the best performance by far [124], compared with other implementations, it still only improves Dice but degrades Hausdorff and ASD. our method, on the other hand, obtains higher performance on all the three metrics on DenseVoxNet. Furthermore, compared with the second-best method [110][124], the average improvement of DenseVoxNet on Dice is 1.2%, while our method can achieve an average improvement of $\sim 1.8\%$ for Myocardium on DenseVoxNet.

We also extended the same evaluations to another state-of-the-art FCN–3D-DSN, to explore the response of our method to different FCN architectures. As shown in Table 3.2, the trend of the results are similar to that of DenseVoxNet, except for lower segmentation

accuracy. Note this is caused by the neural network structure difference, and DenseVoxNet currently achieves the state-of-the-art segmentation performance. As expected, again, our method significantly outperforms JPEG-2000 at the same compression rate ($30\times$) across all the three metrics, i.e. 0.013 (myoscardium) and 0.007 (blood pool) on average for dice score, while providing almost the same segmentation performance as that of uncompressed version–original ($1\times$). These results clearly show the generalization of our method.

It is also notable that from both tables, the segmentation results from compressed images using our method sometimes even outperform that of original images. This is because compression as frequency-domain filtering also has denoising property. Although the training process attempts to learn comprehensive features, the importance of the same frequency feature may vary from one image to another for a trained DNN. As a result, after compression, the segmentation accuracy of some images may be improved because the unnecessary features that can mislead the segmentation are filtered, as demonstrated in Fig. 3.7(b) (Our method is better than Original CT). For most images, the segmentation accuracy after compression is still slightly degraded compared with the original images due to minor information loss at high compression rates, though our compression method tries to minimize the loss of important features.

3.3.4 Comparison of Compression Rate

In this section, we explore to what extent our proposed machine vision-oriented compression framework can improve the compression with regard to the human-visual based 3D JPEG-2000, for medical image segmentation. For a fair compression, we compared the compression rate (CR) of these two methods under the same segmentation accuracy for myocardium using DenseVoxNet. Dice score (0.834) was selected as it is the prime metric to measure the quality of image segmentation. Since the compression rate may vary

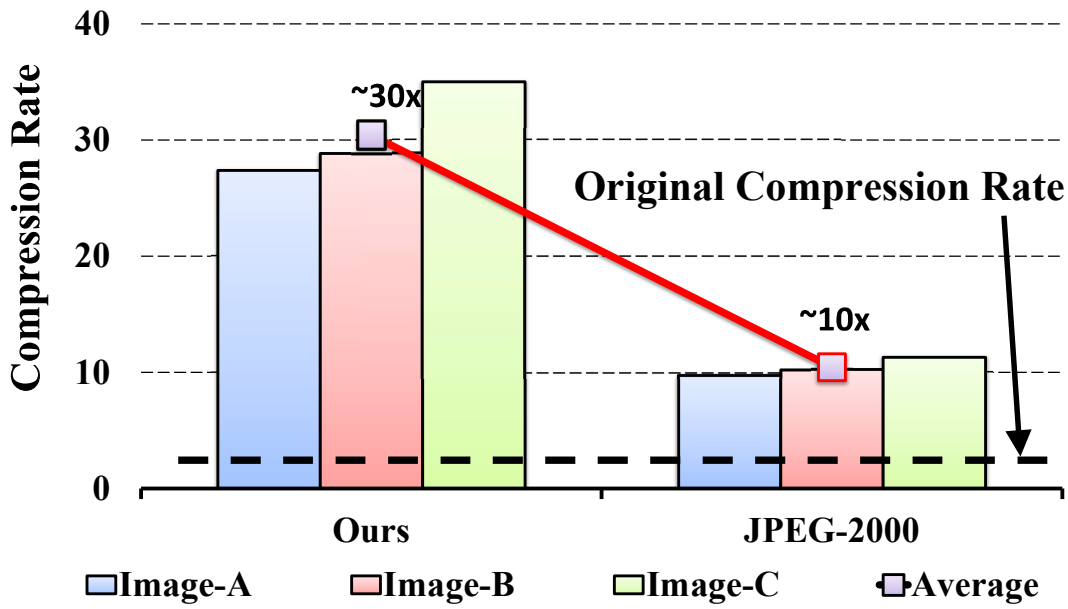

Figure 3.6: Compression rate comparison of our method v.s. JPEG-2000 under the same segmentation accuracy.

from one image to another, we chose three representative images from the dataset. As Fig. 4.3 shows, our method can always deliver the highest compression rate across all the images. On average, it achieves $30\times$ compression rate over the original uncompressed image. Compared with 3D-JPEG 2000, our method can still achieve $3\times$ higher image size reduction, without degrading the segmentation quality. Still taking the example from Chapter 1, we assume the transmission time of a 3D CT image of size 300MB via fixed broadband internet ($22.79Mb$) to cloud is 13s, while the image segmentation computation time on cloud is merely 100ms. Putting these two together, a single image segmentation service time on cloud for our method ($30\times$) and JPEG-2000 ($10\times$), are 0.53s and 1.4s, respectively, translating into $2.6\times$ speed up.

3.3.5 Visual results

The results for four randomly selected slices are shown in Fig. 3.7. From the figure we can see that quite significant differences exist between the segmentation results from the

original image and the one compressed by JPEG-2000, though visually little distortions exist between the two.

The results may seem surprising at first glance, but it is also fully justifiable. The boundaries in medical images mainly contribute to the high frequency details, which cannot be perceived by human eyes. As such, existing compression techniques will ignore them while still attaining excellent compression quality. Yet these details are critical features that neural networks need to extract to accurately segment an image. Similarly, many low frequency features in a medical image such as brightness of a region are important for human vision guided compression, but not at all for segmentation. In other words, human vision and machine vision are completely different with regard to the segmentation task.

3.3.6 Overhead

Our method is built upon 3D-JPEG 2000 by only adding two simple operations: standard deviation calculation for 16 subbands and equation set solution (Equation (3.4)) with only four variables. Since we reuse the majority of JPEG-2000's function units, the compression and decompression time are at the same level as that of JPEG-2000, e.g., 0.12ms for a 512$\times$512 image [72], which is almost negligible compared with image transmission and segmentation time. Therefore, we expect that our light-weighted machine vision guided 3D image compression framework can find broad applications in medical image analysis.

3.4 Conclusion

Due to the high computation complexity of DNNs and the increasingly large volume of medical images, cloud based medical image segmentation has become popular recently. Medical image transmission from local to clouds is the bottleneck for such a service, as it is much more time-consuming than neural network processing on clouds. Although

there exist a lot of 3D image compression methods to reduce the size of medical image being transmitted to cloud hence the transmission latency, almost all of them are based on human vision which is not optimized for neural network, or rather, machine vision. In this chapter, we first present our observation that machine vision is different from human vision. Then we develop a low cost machine vision guided 3D image compression framework dedicated to DNN-based image segmentation by taking advantage of such differences between human vision and DNN. Extensive experiments on widely adopted segmentation DNNs with HVSMR 2016 challenge dataset show that our method significantly beats existing 3D JPEG-2000 in terms of segmentation accuracy and compression rate.

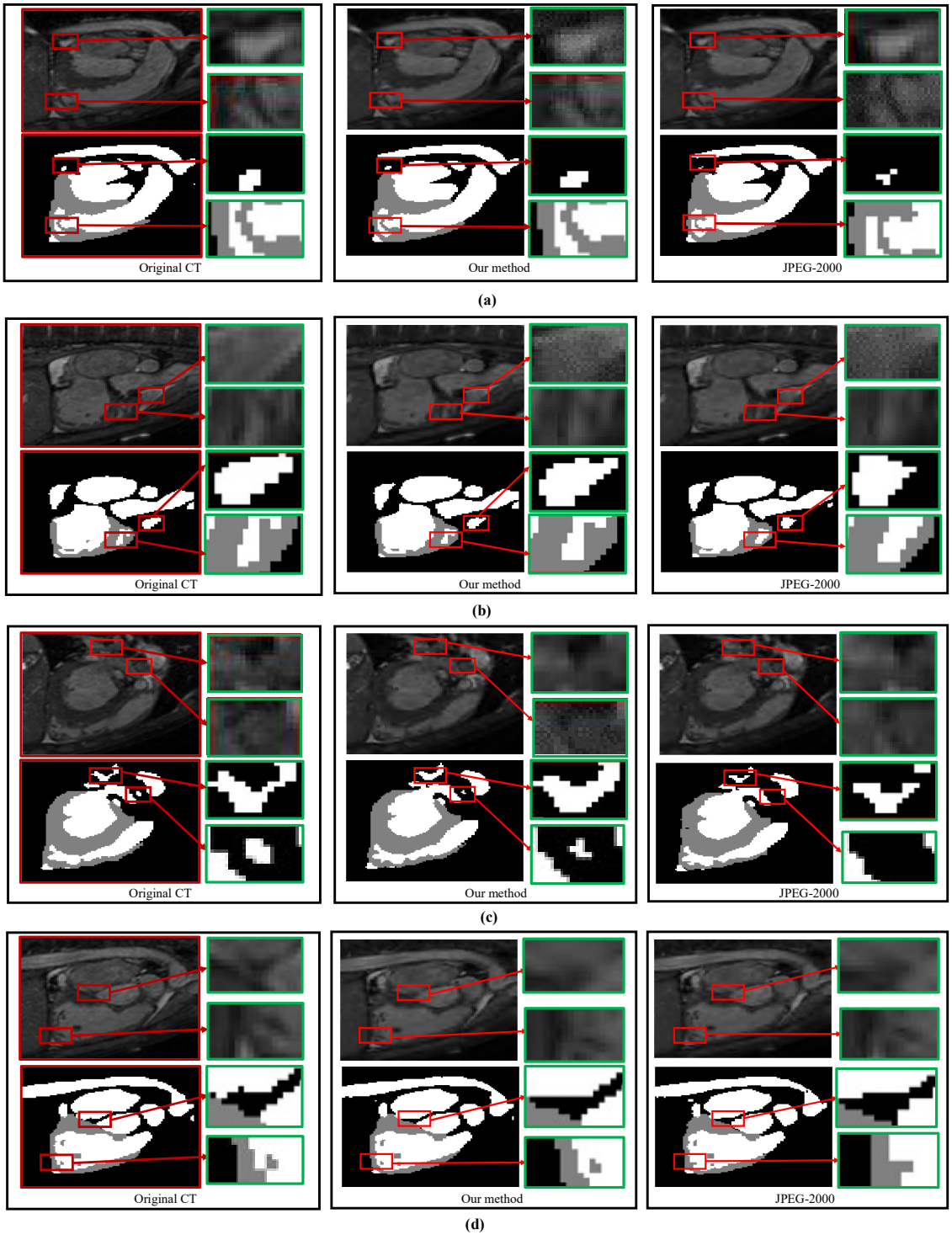

Figure 3.7: Segmentation details of four slices in a CT image in HVSRMR 2016 Challenge dataset [78], compressed using our method and JPEG-2000, and segmented by DenseVoxNet [124]. Many details are missing in the segmentation results from JPEG-2000 compressed images but not in our method. Quantitative comparisons can be found in Section 3.3.

CHAPTER 4

ORCHESTRATING MEDICAL IMAGE COMPRESSION AND SEGMENTATION NETWORKS FOR EFFICIENT TRANSMISSION AND ACCURATE SEGMENTATION IN THE CLOUDS

Deep learning based medical image segmentation in cloud offers outstanding segmentation performance thanks to recent model innovation and computing hardware acceleration. However, one major factor that limits its overall service speed is the long image data transmission latency from local to cloud, which could far exceed the segmentation computation time in cloud. Existing image compression techniques are unable to achieve a sufficient compression rate to dramatically reduce the data offloading overhead which dominates the whole service time, while maintaining the same level of segmentation accuracy. This is because they are all developed upon human visual system, whose image perception pattern could be fundamentally different from that of deep learning-based image segmentation. Motivated by this key observation, in this chapter, we propose a generative segmentation architecture consisting of a compressive auto-encoder, a segmentation network and a discriminator network. Our design synthetically considers both segmentation and compression, and orchestrates the different structures and loss functions, for improving segmentation accuracy and efficiency simultaneously. Our results show that proposed architecture can provide much better compression rate and segmentation accuracy than the-state-of-the-art compression techniques, translating into great improvement on the cloud-based medical image processing efficiency.

In this work we propose to orchestrate medical image compression and segmentation networks for efficient transmission and accurate segmentation in the clouds. Particularly, our end-to-end method trains several neural networks simultaneously for both image compression locally and segmentation in the cloud using adversarial learning, thus to make the two steps matched to extract and retain the most important features for neural network

segmentation. The neural network for image compression is designed to be light-weighted, which fits well for local processing. We conducted comprehensive experiments via Pytorch framework on several 2D and 3D segmentation networks using both ISIC 2017 (2D) [25] and HVSMR 2016 Challenge datasets (3D) [78].

The main contributions of our work are as follows:

1. We propose a framework which integrates compressive auto-encoder and generative segmentation network (with discriminator network), so as to fully unleash the compression potential of auto-encoder by truly embracing the concept of “machine vision"–compression by the direct guidance of enhanced segmentation network.
2. We design a series of training loss functions to optimize the compressive segmentation in the proposed architecture, for achieving high compression rate while maintaining the segmentation accuracy.
3. We conduct comprehensive evaluations on 2D and 3D medical images. Results show our design can improve the compression rate by 2 orders-of-magnitude comparing with the uncompressed images, and increase the segmentation accuracy remarkably over the state-of-the-art solutions.

4.1 Background and Related Work

4.1.1 Medical Image Segmentation

Medical image segmentation has always been one of the most important tasks in medical imaging research. It extracts different tissues, organs, pathologies, and biological structures, to support medical diagnosis, surgical planning and treatments. Recently, deep neural networks (DNNs), particularly fully convolutional networks (FCNs), boost the performance of medical image segmentation and outperform the previous methods by a large margin.

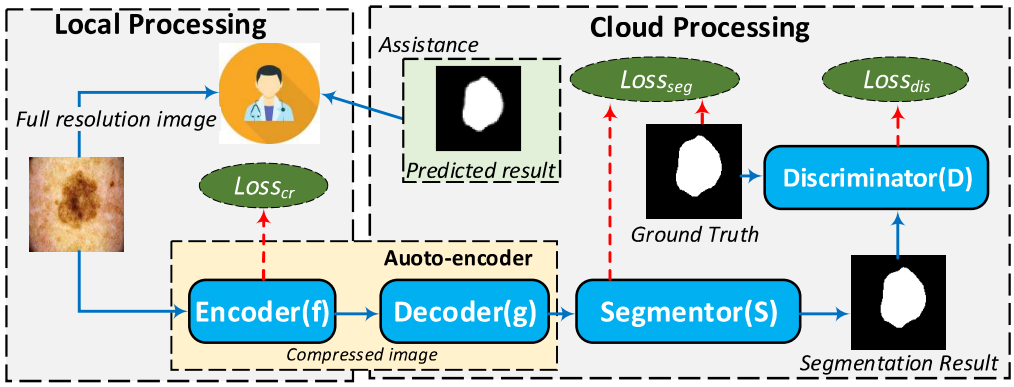

Figure 4.1: Our orchestrating medical image compression and segmentation networks design flow.

Ronneberger et al. [84] proposed the U-Net, a U-shaped deep convolutional network that adds a symmetric expanding path to enable precise localization. The DCAN model by Chen et al. [20, 21] added a unified multi-task object to the U-Net learning framework to improve the accuracy of boundary detection. There are also some FCNs for 3D images, such as 3D U-net [24] and V-net [73], which adopt 3D convolution for processing.

4.1.2 Medical Image Compression

Traditional image compression standards such as JPEG [109] and JPEG-2000 [13, 14] are widely adopted in medical image compression, They usually employ Discrete Cosine Transform (DCT) and Discrete Wavelet Transform (DWT) for compression while preserving important visual information for humans. A series of alternative methods based on them, are also proposed to further improve the performance of medical image compression. For example, in [121], authors improved the compression performance for medical diagnosis by eliminating the unimportant image regions. In [69], a 3D block classification is proposed to conduct compression at the volume level.

Besides aforementioned traditional approaches, image compression using trainable auto-encoders which are built upon neural networks, have recently received great attention.

[74] developed a joint auto regressive and hierarchical method which includes a normal auto-encoder to first create compressed representation, followed by a trainable sub-network dedicated to entropy coding for further compression. The goal allies with the classic methods like JPEG/JPEG2000, which attempt to achieve a high compression rate without degrading human perceived image quality, e.g. high PSNR, SSIM. In [105], authors proposed to directly train an DNN with compressed representation produced by an auto-encoder, so as to reduce the overhead of data transmission and DNN computation. To balance the accuracy and compression rate, the auto-encoder needs to keep sufficient details of input images, and thus is trained by minimizing the mean square error (MSE) of the original and reconstructed image. [4] also developed a generative adversarial network for image compression by combing the auto-encoder with a discriminator to improve image visual quality and compression rate. Similar as [105], the auto-encoder is trained by minimizing the visual difference between original and reconstructed images.

Apparently, all these classic and auto-encoder solutions compress images *under the guidance of human perceived image quality, which could be fundamentally different from that of neural network-based image segmentation and lead to limited compression efficiency under the context of machine vision.* While the idea of using generative adversarial network in our work seems to be similar to [105], the training of auto-encoder is very different: our work adopts a joint training process guided by segmentator and descriminator with different loss functions applied to the encoder and decoder, respectively, while that of [105] is only assisted by the discriminator with human-visual quality as a measurement.

Very recently [66] proposed a proof-of-concept “machine vision" guided 3D image compression framework based on JPEG-2000 to improve the accuracy and efficiency of 3D CT image segmentation. While we adopt the similar design concept in this work, their basic approach is to slightly modify quantization steps of standard JPEG-2000, so as to achieve $\sim 2\times$ compression ratio than the original JPEG-2000 without harming the

segmentation quality. As a result, its efficiency is still limited because of the underlying JPEG-2000 structure and heuristic quantization step searching. Moreover, this solution only focuses on the compression itself and does not jointly optimize the compression and segmentation. In contrast, *this work for the first time formulates a framework which jointly takes the auto-encoder based compression loss, segmentation loss and discrimination loss into consideration through an adversarial training manner, and can compress images in a way that better assists neural network-based image segmentation ("machine vision"), thereby achieving much higher compression rate and better segmentation accuracy.*

4.2 Our Methodology

Fig. 5.1 depicts an overview of our framework, which consists of three integrated components: the auto-encoder (C), segmentation network (S) and the discriminator network (D). Specifically, **1)** C functions as a lossy image compression/decompression engine, with the encoder producing highly compressed data locally and decoder recovering image data at the cloud for segmentation after receiving the compressed bits through a wireless link. As such, the network of auto-encoder, especially encoder for compression in local, should be light-weighted for fast processing and low cost; **2)** The segmentor S is a fully convolutional neural network (FCN) that can generate a probability label map from the input image reconstructed by the decoder of C; **3)** The discriminator D aims to capture any difference between the predicted label map from S and the corresponding ground truth label map. C, S and D are alternatively trained in an adversarial fashion with the goal of solving a min-max optimization problem: the training of C and S aims to minimize the label feature loss while maximizing the compression rate (CR), while the training of D attempts to maximize this loss. The goal is to achieve a high enough compression rate, under the guidance of DNN-based segmentation quality measurement (NOT human perceived

image quality, e.g. PSNR), so as to significantly accelerate the speed of cloud-based image segmentation service while providing similar or even higher segmentation quality at the testing stage. As we shall show later, images after our auto-encoder, though presenting lower visual quality because of only keeping most important features for segmentation (see Fig. 4.4), actually help the segmentation from two aspects: a much higher compression rate (faster service speed) with a similar accuracy, or better accuracy for a similar compression rate, when compared with existing solutions.

Note images sent to the cloud will be only used by segmentation networks in order to generate accurate segmentation label maps. Such predicted label maps, together with the local stored high-resolution image copies, can assist doctors [59] for better medical diagnosis, surgical planning/treatment etc (see Fig. 5.1). Therefore, these images do not necessarily preserve high visual quality, but can be compressed as much as possible (for short service latency) as long as the segmentation results are good.

4.2.1 Architecture Design

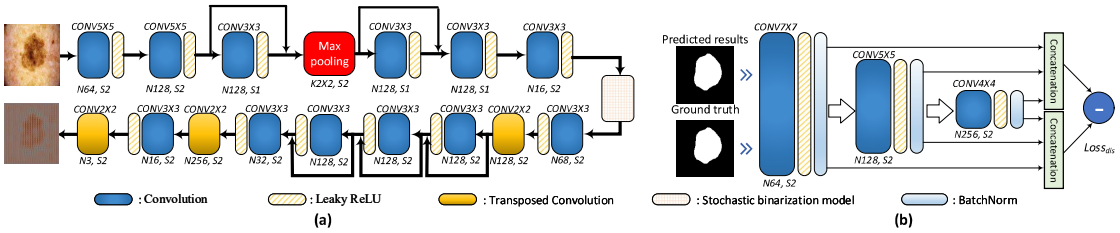

Figure 4.2: Illustration of (a) auto-encoder; (b) discriminator in our design.

Auto-encoder C: In our design, the auto-encoder (C) is followed by the segmentor (S), with the output of C feeding into S for segmentation accuracy feedback. An auto-encoder consists of an encoder f (compression), decoder g (decompression) and a probabilistic model Q (bit number estimation for encoder output representation, e.g. entropy coding). Unlike existing auto-encoders which attempt to minimize the pixel-wise visual distortions

between original image and reconstructed image, our auto-encoder focuses on minimizing the difference between the predict label from S and ground truth label y for an input x–$d(S(C(x)), y)$, as well as the number of bits needed after the encoder f–$-log_2Q(f(x))$ for the highest compression rate, during the training process. This indicates that our auto-encoder is dedicated to compressing images in an DNN-favorable manner for segmentation purpose.

During the implementation, the computation cost and latency of the auto-encoder, especially the encoder, should be low enough for local processing, e.g. comparable with existing light-weighted JPEG/JPEG-2000 and much lower than the complex segmentation network in cloud. Fig. 4.2(a) shows the detailed structure of our example 2D auto-encoder inspired by [103]. It only consists of 6 (2) and 9 (3) convolution layers (residual blocks) for encoder and decoder, respectively. Such an unbalanced structure can further reduce the encoder's local computing cost and latency. The downsampling is performed by convolution with a stride 2 and maxpooling, while upsampling is realized by transposed convolution operation. The 3D auto-encoder can be also designed following a similar approach. Table 4.1 further compares the detailed parameters among encoder, decoder and a segmentor [122] which we will use as a baseline in evaluation. Among them, the encoder has the lowest cost, e.g. only $1/179$, $1/11$ and $1/7$ of the number of parameters, number of covolutional layers and ReLU layers in segmentor. As we shall show in Table 4.4, this translates into significant low processing time in local.

Semgentor S and Predict-oriented discriminator D: We do not design new seg-

Table 4.1: The comparison of detailed settings for encoder, decoder and segmentor.

	Encoder	**Decoder**	**Segmentor**	**Times**
CONV layers	6	9	66	11
Normal	0	0	17	N/A
ReLU	6	7	42	7
# Parameters	872465	1308722	156265344	179

mentors for our framework, instead, we adopt existing representative networks for 2D and 3D segmentation tasks [122, 50], and demonstrate the scalablity of our design. To further compensate the potential accuracy loss because of the joint training of S and C, our framework incorporates a discriminator D after the S [68, 122]. Fig. 4.2(b) shows the detailed structure of D. The inputs of D are ground truth label map and predicted label map from S, the output from each layer will concatenate together first and then their difference will be used as the label feature loss to train D.

4.2.2 Training Loss Design

To train the framework, we now design the loss function dedicated to each network. Given a dataset with N training images x_n, and y_n as the corresponding ground truth label map, the multi-scale label feature loss ($loss_{dis}$) and segmentation loss ($loss_{seg}$) can be defined as follows:

$$\begin{aligned} loss_{dis} = \min_{\theta_C,\theta_S} \max_{\theta_D} \zeta(\theta_C, \theta_S, \theta_D) = \\ \frac{1}{N}\sum_{n=1}^{N} \ell_{mae}(\phi_D(\phi_S(\phi_C(x_n))), \phi_D(y_n)) \end{aligned} \tag{4.1}$$

$$loss_{seg} = \min_{\theta_C,\theta_S} \xi(\theta_C, \theta_S) = \ell_{mse}(\phi_S(\phi_C(x_n)), y_n) \tag{4.2}$$

where θ_C, θ_S and θ_D are weight parameters of C, S and D, respectively. ℓ_{mae} is the mean absolute error or L_1 distance inspired from [122]–$\phi_S(\phi_C(x_n))$ is the prediction result of S after input x_n is compressed by auto-encoder C and $\phi_D(\cdot)$ represents the multi-scale hierarchical features extracted from each convolutional layer in D. ℓ_{mse} is the MSE between predicted label from S and ground truth label. $\phi_C(\cdot)$, $\phi_S(\cdot)$ and $\phi_D(\cdot)$ (i.e. the hierarchical features extracted) represent the functions of C, S and D, respectively. We model the loss for the discriminator as:

$$-loss_{dis} = -\min_{\theta_C,\theta_S} \max_{\theta_D} \zeta(\theta_C, \theta_S, \theta_D) \tag{4.3}$$

We set this loss with a negative value to maximize the difference between the predicted label and ground truth label. On the contrary, we add the reserved version of this loss (positive value) to C and S, with the goal of minimizing such loss for the combined C and S. Therefore, the total loss for segmentor and auto-encoder is:

$$\begin{aligned} loss_{total} = loss_{dis} + loss_{seg} = \\ \min_{\theta_C,\theta_S} \xi(\theta_C,\theta_S) + \min_{\theta_C,\theta_S} \max_{\theta_D} \zeta(\theta_C,\theta_S,\theta_D) \end{aligned} \tag{4.4}$$

Finally, we introduce a compression loss ($loss_{cr}$) to optimize the output of encoder for achieving the best compression rate. We assume the encoder/decoder of the auto-encoder is f/g, and use e to estimate the number of bits for the representation after f, e.g. entropy coding. Since this coding process is non-differentiable, we employ a continuous differentiable Jensen's inequality [102] to estimate the upper bound of the number of needed bits. This estimation is used to train the encoder [103]. Note this loss is only sent to the encoder in auto-encoder without involving the decoder. Then the total loss for the encoder f of auto-encoder C is:

$$\begin{aligned} loss_{cr} + loss_{seg} + loss_{dis} = \underbrace{min(e(f(x_n)))}_{No.bits} \\ \underbrace{+ \min_{\theta_C,\theta_S} \max_{\theta_D} \zeta(\theta_C,\theta_S,\theta_D) + \min_{\theta_C,\theta_S} \xi(\theta_C,\theta_S)}_{Seg.distortion} \end{aligned} \tag{4.5}$$

4.2.3 Training and Testing

We train our framework by following an alternating fashion: For each training epoch, **first**, we fix the parameters of D and only train that of C and S for one step using above designed loss functions, e.g. $loss_{total}$ (Eq. 4.4) for the decoder of C and segmentor S, and $loss_{cr} + loss_{total}$ (Eq. 4.5) for the encoder of C for optimized compression rate of encoded data from C; **Second**, we fix the parameters of C and S and train D by the gradients computed from its loss function ($loss_{dis}$). As Eq. 4.1 shows, this training process

behaves more like a min-max game: while C and S try to minimize $loss_{dis}$, D attempts to maximize it. As a result, the training gradually improves the segmentation results of C, S and D, as well as the compression efficiency of C after each epoch until reaching the convergence.

At the testing process, only C and S are used to predict the segmentation accuracy and D is not involved. The input image is first sent to the encoder of auto-encoder C, then a stochastic binarization algorithm will be applied to the encoded data, i.e. the encoded representation is in binary format. After that, the binary data is further encoded by entropy coding. We adopt the entropy rate estimation method in [103] to estimate the final number of bits. At the decoding process, the encoded binary data is directly decoded by the decoder of auto-encoder C to reconstruct the image. Then it is sent to the segmentation network S for the label map prediction.

4.3 Evaluation

4.3.1 Experiment Setup

Our proposed method is built upon Pytorch [79] framework on a server with 6 cores i7-6850K CPU and multiple 2560 CUDA cores GTX 1080 GPUs. The encoder (for compression) is implemented on both GPU and less powerful CPU to validate its feasibility for local low-cost processing. The whole network is realized by heavily modifying the adversarial segmentation network proposed by [122] with the integration of our auto-encoder architecture. **Benchmarks.** Two datasets are selected for both 2D and 3D image segmentation tasks. For 2D benchmark, ISIC 2017 challenge dataset [25] is adopted. This fully annotated dataset provides 2000 training images, 150 validation images and 600 testing images for the Lesion segmentation task. For 3D benchmark, we use the HVSMR

2016 challenge dataset [78], which consists of in total 5 3D cardiac MR scans for training and 5 scans for testing. Each image includes three segmentation labels: myocardium, blood pool, and background. Note that directly using the large-size 3D medical images (e.g. 300MB) to train neural networks is difficult. Thus we randomly crop the original CT image to many smaller pieces of data to facilitate training and overcome the overfitting, which is consistent with [66].

Metrics. We evaluate our methodology from four aspects: **segmentation performance, compression efficiency, cloud-based service latency and visual analysis.** To measure the segmentation performance, we use the widely adopted Intersection over Union (IoU) and Dice coefficient (Dice) as the two metrics. The higher IoU/Dice score indicates better agreement (or accuracy) between the segmentation result and the ground truth. To evaluate the proposed auto-encoder based compression, we use bits per pixel (bpp) as the index. For the same image, the lower bpp means better compression efficiency or higher compression rate. Note we also estimate **cloud-based service latency**, composed of transmission and segmentation time, by considering the impact of compression/decompression. Visual analysis illustrates the visual quality of the compressive auto-encoder reconstructed images (predict labels) and original images (ground truth labels).

Auto-encoder, Segmentor & Discriminator Settings. The structure of 2D auto-encoder is shown in Fig. 4.2(a). The 3D auto-encoder is as follows: the encoder consists of 4 convolutional layers for 3D downsampling by 3D convolutional operation with a $3 \times 3 \times 3$ kernel and stride 2. The decoder performs 3D upsampling implemented by 4 3D Transpose operation with a $2 \times 2 \times 2$ kernel and stride 1, through 8 convolutional layers for image reconstruction. For both 2D and 3D auto-encoders, the stochastic binarization [104] is applied to binarize the created code (compressed data) for achieving a high compression rate. For segmentation networks, we choose FCN32s [67] (i.e. upsampling stride 32 predictions back to pixels in a single step without combining encoder layer) and FCN16s [67] (i.e.

combining predictions from both the final layer and the pool4 layer at stride 16), the default segmentation network in original SegAN (i.e. UNet) and 3D-UNet [50]. The discriminator network (see Fig. 4.2 (b)) is developed based on [122], including 3 convolutional layers along with batch normalization and leaky ReLU layers with the feature maps from each hierarchical layer concatenated together to calculate $loss_{dis}$.

Evaluated Designs. We use the recent SegAN framework [122], which includes both segmentation and discriminator networks, as the baseline. Then we modify it and generate several designs with different components, e.g. incorporating the auto-encoder into it or replacing its segmentation network, for comprehensive evaluations:

- **Our (Seg+Dis).** Design with segmentator and discriminator but no auto-encoder.
- **Our (Auto+Seg).** Design with auto-encoder and segmentator but no discriminator;
- **Our (Auto+Seg+Dis). Standard** design with auto-encoder, segmentator and discriminator, without considering the compression loss ($Loss_{cr}$). This design is expected to offer the best segmentation accuracy, but limited compression efficiency.
- **Our (Auto+Seg+Dis+CR). Enhanced** design with auto-encoder, segmentator, discriminator and compression loss ($Loss_{cr}$). This design should achieve aggressive compression with slightly degraded segmentation accuracy.

Besides, we select standard JPEG-2000 [14, 13], and latest “machine vision" based compression “ [66]", in our evaluation. The auto-encoder and segmentation joint design which is widely adopted by existing works, namely “Auto(MSE)/Seg" here, is also implemented. Here the auto-encoder is trained by MSE of reconstructed image and original image and the reconstructed images are used to train segmentor. As such, we expect “Auto(MSE)/Seg" should suffer from prominent segmentation accuracy loss at high compression rates.

4.3.2 Segmentation Performance

To show the scalability of our proposed design, we evaluate both 2D- and 3D- segmentation on a series of baselines with different segmentation networks and structures.

2D Segmentation. Table 4.2 evaluates the Dice/IoU score of 2D segmentation on several selected designs. For each baseline, we test both segmentation networks–FCN16s and FCN32s. As shown in Table 4.2, our designs always achieve higher Dice and IoU scores than SegAN on all testing cases. In particular, without compression, "Our (Seg+Dis)" improves both Dice (0.809) and IoU (0.715) by ~ 0.01 than that of SegAN (0.798 on Dice and 0.706 on IoU) with FCN32s, due to the optimized predict-oriented discriminator in our design. *We would like to emphasize that the achieved performance improvement of our method is very significant for biomedical segmentation tasks* [110, 124]. With integrated auto-encoder, "Our (Auto+Seg+Dis)" achieves the best segmentation accuracy (0.813 on Dice and 0.715 on IoU) among three designs on FCN32s. The similar trend can be also observed on FCN16s– an enhanced segmentation network compared with FCN32s, thus achieving better results on all selected cases. This result indicates **our auto-encoder based compression, instead of degrading the accuracy, can sometimes even further improve the segmentation performance.** This is because proper compression can introduce the denoising effect. Although the training process attempts to learn as many features as possible, the necessary features for segmentation needed by each image may vary from one to another. Therefore, after compression, the segmentation accuracy of some images can be improved because of removing unnecessary features that may confuse the segmentation (we will present and discuss the compressed image samples in "Visual

Table 4.2: 2D segmentation results on ISIC 2017 dataset.

	FCN32s			FCN16s		
	Our (Auto+Seg+Dis)	**Our (Seg+Dis)**	**SegAN**	**Our (Auto+Seg+Dis)**	**Our (Seg+Dis)**	**SegAN**
Dice	**0.813**	0.809	0.798	**0.814**	0.812	0.805
IoU	**0.715**	0.715	0.706	**0.718**	0.717	0.708

Table 4.3: 3D segmentation results on HVSMR 2016 challenge dataset [78]. Our: Our (Auto+Seg+Dis)

	Myocardium						Blood Pool					
	Dice			IoU			Dice			IoU		
	Uncomp.	[66]	Our	Uncomp.	[66]	Our	Uncomp.	[66]	Our	Uncomp.	[66]	Our
Img.1	0.895	0.868	0.899	0.809	0.756	0.817	0.915	0.876	0.927	0.844	0.818	0.864
Img.2	0.829	0.798	0.831	0.708	0.681	0.711	0.951	0.903	0.948	0.916	0.875	0.921
Img.3	0.811	0.782	0.815	0.672	0.652	0.674	0.883	0.858	0.888	0.807	0.754	0.808
Img.4	0.877	0.853	0.874	0.780	0.758	0.776	0.955	0.906	0.956	0.913	0.872	0.915
Img.5	0.809	0.778	0.810	0.679	0.647	0.681	0.883	0.849	0.881	0.806	0.779	0.788
Average	0.844	0.816	**0.846**	0.729	0.699	**0.732**	0.918	0.878	**0.920**	0.857	0.820	**0.859**
bpp	Uncompressed ($\sim$1.1)			[66]($\sim$0.04)			**Our($\sim$0.014)**					

Analysis").

3D Segmentation. Table 4.3 shows our 3D segmentation result on HVSMR 2016 dataset using 3D-UNet [50] as the segmentation network. We test 5 3D CT image volumes with segmentation targets “Myocardium” and “Blood Pool”, and compare the Dice/IoU scores among “Our (Auto+Seg+Dis)”, “ [66]” (optimized JPEG-2000) and the uncompressed design. For a fair comparison, the compression rate in “ [66]” and “Our (Auto+Seg+Dis)” is maintained at the same level (bpp = 0.014). Compared with the uncompressed image segmentation, our design improves the average Dice (IoU) score by 0.002 (0.003) and 0.002 (0.002) on “Myocardium” and “Blood Pool”, respectively, which is similar to the 2D segmentation. However, compared to the state-of-the-art “machine vision" guided compression “ [66]", our design achieves more significant improvement on 3D segmentation, i.e., the average Dice (IoU) score is increased by 0.03 (0.033) and 0.042 (0.039) on “Myocardium” and “Blood Pool”, respectively. As expected, “ [66]" cannot keep high accuracy at a high compression rate ($\sim 78\times$), e.g. much lower than that of the uncompressed one. These results show great scalability and outstanding segmentation performance of our design for 3D images.

Segmentation under different architectures. Figure 4.3(a) further compares the segmentation performance of 2D dataset of selected baseline designs under different architectures and component combinations. We use the same segmentation network as that

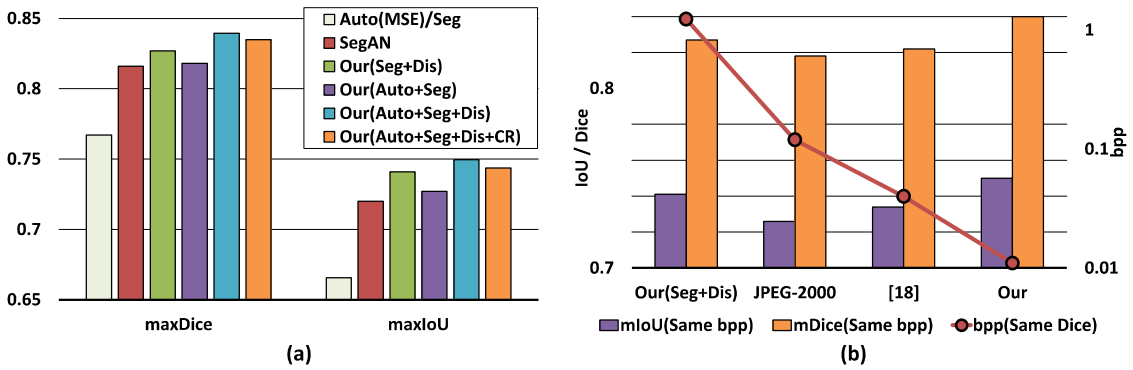

Figure 4.3: (a) Segmentation results under various combinations. (b) Segmentation accuracy/bpp comparison with prior methods. Same bpp (bars): Our (Auto+Seg+Dis); Same dice (line): Our (Auto+Seg+Dis+CR).

of default SegAN, but replace its descriminator or add our auto-encoder, to develop these designs. Both Dice and IoU exhibit a similar trend on all selected designs. We use Dice as an example to analyze the results. Compared with SegAN, “Our (Auto+Seg)" can slightly improve Dice score with filtered features after compression. Moreover, “Our (Seg+Dis)" achieves higher Dice score than “Our (Auto+Seg)”. These results indicate the proposed predict-oriented discriminator can better improve the segmentation accuracy with the combination of $Loss_{seg}$ and $Loss_{dis}$ than that of auto-encoder based compression. In particular, our standard design “Our (Auto+Seg+Dis)" shows the best Dice (and IoU) among all schemes. However, by further applying compression loss $Loss_{cr}$, the segmentation accuracy of “Our (Auto+Seg+Dis+CR)" is slightly degraded on both Dice and IoU. This is because the applied compression loss performs more aggressive compression (i.e., $> 1.5\times$) on “Our (Auto+Seg+Dis+CR)” than that of “Our (Auto+Seg+Dis)", leading to slightly sacrificed segmentation accuracy. On the other hand, as expected, “Auto(MSE)/Seg", an auto-encoder and segmentor joint design adopted by existing works and guided by human visual quality loss (e.g. MSE), achieves the lowest Dice/IoU among all designs at the same level of bpp (e.g. 0.013).

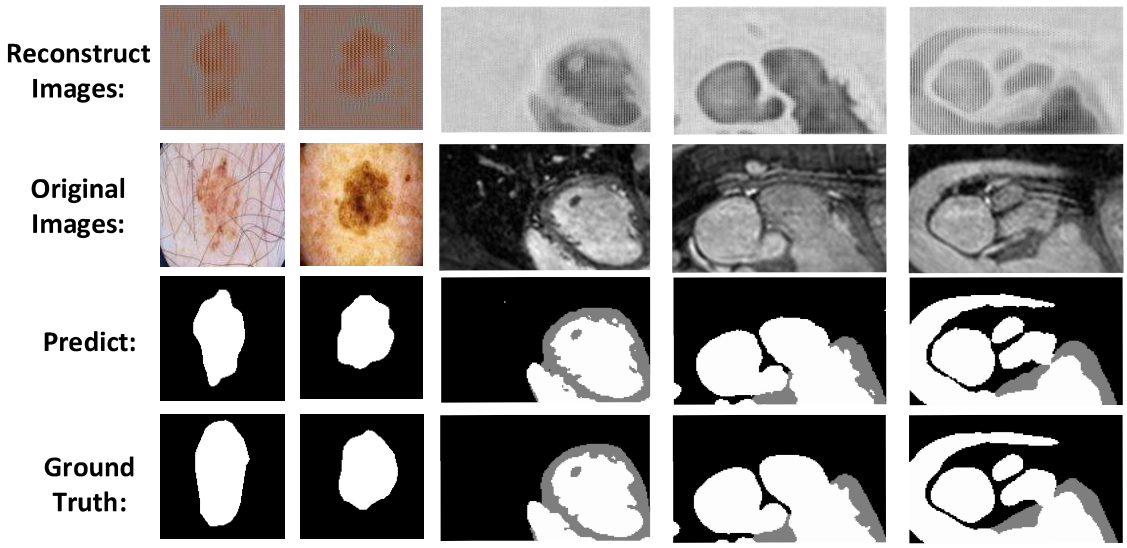

Figure 4.4: Comparison between original and reconstructed (decompressed) images from auto-encoder of 2D RGB images (left 2 columns) and 3D cardiovascular magnetic resonance (CMR) images (right 3 columns) with corresponding predict label and ground truth label.

4.3.3 Compression Efficiency

To better evaluate the compression performance, we consider both compression (bpp) and segmentation (IoU/Dice), and compare the results of proposed architecture with four different compression approaches, including the uncompressed “Our (Seg+Dis)", the auto-encoder based “Our (Auto+Seg+Dis+CR)", the standard “JPEG-2000" and the-state-of-the-art machine-vision based compression (best) from “ [66]". Figure 4.3(b) reports the results. As shown by the bars, the uncompressed “Our (Seg+Dis)" shows slightly better (IoU and Dice scores than “JPEG-2000” and “ [66]", since these two existing approaches cannot well preserve the necessary features during compression. Instead, “Our (Auto+Seg+Dis)" with our auto-encoder can always achieve the best segmentation performance for both IoU and Dice, by better keeping important features during compression. Figure 4.3(b) (the line) further shows the average bpp of each compression approach characterized at the same level of segmentation accuracy (i.e., Dice = 0.84). Under such a constraint, “Our (Auto+Seg+Dis+CR)” achieves the best compression rate (average bpp = 0.013) among

all approaches, e.g. improving compression rate by almost two orders of magnitude than that of uncompressed images (average bpp = 1.24), by an order of magnitude than that of “JPEG-2000" (average bpp = 0.12), and $> 3\times$ than that of “ [66]" (average bpp = 0.04). Such significant improvement can be attributed to two factors: the auto-encoder design and the compression loss. The code size (i.e., compressed data) in our implemented auto-encoder is $16 \times 8 \times 8$ (i.e. 1024 bits), while the input image is $3 \times 128 \times 128$ (i.e. 49152 pixels), translating into the baseline bpp– ~ 0.02. Later, the upper bound of number of bits is adjusted by the estimated entropy coding e. Therefore, the actual bpp (0.013) is further reduced by $\sim 1.54\times$ with proposed compression loss $Loss_{cr}$.

4.3.4 Cloud-based Service Latency

We evaluate the processing efficiency of cloud-based image segmentation under the same quality by using different compression techniques. We assume the transmission latency of an uncompressed CMR image (300MB) to cloud via an ideal network (e.g. stable fixed broadband internet 22.79Mbps without bandwidth contention) is 13s, which is much larger than the segmentation computation time ($\sim 100ms$) in the cloud. Besides, the compression/decompression time overhead incurred by auto-encoder should be also considered. Table 4.4 further shows the detailed breakdown of image transmission time, encoder (compression) time, decoder (decompression) time and segmentation time for four designs. The image compression/decompression time overhead of “JPEG", “ [66]”,

Table 4.4: Time Complexity Comparison.

	Image (MB)	**Trans.(s)**	**Enc.(s)**	**Dec.(s)**	**Seg(s)**	**Total (s)**
Uncomp.	300	13	0	0	0.1	13.1
JPEG	30	1.4	0.0013	0.0013	0.1	1.5026
“ [66]”	10	0.53	0.0015	0.0015	0.1	0.633
Our(Auto+Seg+Dis)(GPU)	3.3	0.16	0.0003	0.0003	0.1	0.2606
Our(Auto+Seg+Dis)(CPU)	3.3	0.16	0.003	0.0003	0.1	0.2633

or our auto-encoder, are all negligible compared with their respective transmission time. Putting all time together, the total service time of a single image segmentation in clouds for “JPEG” ($10\times$) and “ [66]” ($30\times$), are 1.5s and 0.63s, respectively. However, our design–“Our (Auto+Seg+Dis)" only takes 0.26s ($50\times$) for a single image segmentation because of the significantly improved compression rate, translating into $5.7\times$ and $\sim 2.4\times$ speedup compared with “JPEG” and latest “ [66]", respectively.

Besides the GPU-based auto-encoder implementation, we further evaluated the time cost of encoder deployed in a less powerful CPU, which is quite common for low-cost local compression, and observed that although the encoding time of auto-encoder in CPU increases by $10\times$ (e.g. 0.003s) comparing with GPU, it is still at the same order of magnitude as JPEG (e.g. 0.0013s). Moreover, the encoding time in CPU is almost negligible when compared with the transmission time (e.g. 0.16s). This means that data transfer time dominates the total service latency, and significantly improving compression rate (far better than JPEG) with our solution is very necessary for service speedup.

Auto-encoder Overhead. We also compare the overhead of our encoder with JPEG compression from a theoretical perspective, since it should be low for local processing. Based on [23], the compression complexity of JPEG can be roughly estimated as $O((kn)^4) + O(N)$, where the first and second term represent the algorithm complexity of DCT and quantization, receptively. Here $n = 8$. k and N denote the total number of 8×8 blocks and pixels in the image, respectively. The encoder complexity of an auto-encoder can be calculated as $O(\sum_{l=1}^{d} = n_{l-1} \cdot s_l^2 \cdot n_l \cdot m_l^2)$ [43], where l is the index of a convolutional layer, d is the number of convolutional layers, n_l/n_{l-1} is the number of filters in the l-th/$l-1$-th layer, s_l is the spatial size of the filter and m_l is the size of the output feature map. Take a $128 \times 128 \times 3$ image and our 2D auto-encoder as an example, the overhead of JPEG compression is $\sim 13\times$ than that of our encoder. This is consistent with our experimental results shown in Table 4.4. Note all reported

time results of compression methods are characterized based on Python implementations.

Deployment Cost. For practical implementation, since each medical imaging task would require a dedicated segmentation network and training process, in our framework, only a local light-weighted auto-encoder needs to be paired with the target cloud network at the training stage for each task and the additional training overhead is very marginal. Moreover, auto-encoder based solution becomes popular in various deep learning tasks. Considering the impressive performance and relatively low overhead, our solution will be very attractive for ever-increasing DNN based medical imaging at cloud.

4.3.5 Visual Analysis

Figure 4.4 compares the visual results of our auto-encoder reconstructed images with original images. The first row represents decompressed images from auto-encoder before feeding into a segmentation network. Compared with original images, they demonstrate lower visual quality for human vision, but can still maintain (or even improve) the segmentation accuracy. These results also indicate that: First, the reconstructed image is not a colorful image. Instead, it only has one red color channel, and the profile is not clear enough for human eyes. Second, some undesired features have been removed. For example, the hairs in the original image of first column are eliminated in the reconstructed image, which actually may make segmentation more accurate. Third, all reconstructed images, regardless of 2D or 3D, are formed by many small blocks (same pattern) and such patterns can further improve compression rate.

4.4 Conclusion

This chapter presents a generative segmentation architecture for compressed biomedical images, which consists of a compressive auto-encoder, a segmentation network and

a discriminator network. We propose to leverage the auto-encoder and different loss function designs to enhance the cloud-based segmentation performance and efficiency by synthetically considering segmentation accuracy and compression rate. We conducted comprehensive evaluations on both 2D RGB and 3D CMR images and compared our design with state-of-the-art solutions. Experimental results show that our design not only significantly improves compression rate, but also increases the segmentation accuracy, outperforming existing solutions by offering better efficiency on cloud-based image segmentation.

CHAPTER 5

FEATURE DISTILLATION: DNN-ORIENTED JPEG COMPRESSION AGAINST ADVERSARIAL EXAMPLES

Image compression-based approaches for defending against the adversarial-example attacks, which threaten the safety use of deep neural networks (DNN), have been investigated recently. However, prior works mainly rely on directly tuning parameters like compression rate, to blindly reduce image features, thereby lacking guarantee on both defense efficiency (i.e. accuracy of polluted images) and classification accuracy of benign images, after applying defense methods. To overcome these limitations, we propose a JPEG-based defensive compression framework, namely “feature distillation", to effectively rectify adversarial examples without impacting classification accuracy on benign data. Our framework significantly escalates the defense efficiency with marginal accuracy reduction using a two-step method: First, we maximize malicious features filtering of adversarial input perturbations by developing defensive quantization in frequency domain of JPEG compression or decompression, guided by a semi-analytical method; Second, we suppress the distortions of benign features to restore classification accuracy through a DNN-oriented quantization refine process. Our experimental results show that proposed “feature distillation" can significantly surpass the latest input-transformation based mitigation such as Quilting and TV Minimization in three aspects, including defense efficiency (improve classification accuracy from $\sim 20\%$ to $\sim 90\%$ on adversarial examples), accuracy of benign images after defense ($\leq 1\%$ accuracy degradation), and processing time per image ($\sim 259\times$ Speedup). Moreover, our solution can also provide the best defense efficiency ($\sim 60\%$ accuracy) against the recent adaptive attack with least accuracy reduction ($\sim 1\%$) on benign images when compared with other input-transformation based defense methods.

In this work, we focus on improving the effectiveness and efficiency of compression based model-agnostic mitigation against adversarial examples. Though standard JPEG

compression has been explored to mitigate the adversarial examples [35, 29], it can neither effectively remove the adversarial perturbations, nor guarantee the classification accuracy on benign images, due to its focus on human visual quality. Instead, we propose the DNN-favorable JPEG compression, namely *"feature distillation"*, by redesigning the standard JPEG compression algorithm, in order to maximize the defense efficiency while assuring the DNN testing accuracy. In specific, 1) We reveal the root reason to limit the JPEG defense efficiency by analyzing the frequency feature distributions of adversarial input perturbations during JPEG compression; 2) Inspired by our observation, we propose a semi-analytical method to guide the defensive quantization process to maximize the effectiveness of filtering adversarial features; 3) We characterize the importance of input features for DNNs by leveraging the statistical frequency component analysis within JPEG, and then develop DNN-oriented quantization method to recover the degraded accuracy (i.e., a side-effect induced by the feature loss in perturbation removal) on benign samples.

Our proposed method is built upon the light modifications of widely adopted JPEG compression and does not require any expensive model retraining or multiple model predictions. Evaluations show that *"feature distillation"* offers significantly improved effectiveness against a variety of mainstream adversarial examples (i.e., $> 90\%$ accuracy on AEs), with very marginal accuracy reduction (i.e., $\leq 1\%$) on benign data. Besides, it well beats recent proposed image transformation based defense like Quilting and TV Minimization in terms of defense efficiency, accuracy and processing speed. Furthermore, our solution offers the best defense efficiency ($\sim 60\%$) with lowest accuracy loss ($\leq 1\%$) against the recent adaptive attack–Backward Pass Differentiable Approximation (BPDA) [8] among existing input-transformation based defenses, though it is not completely immune to such attack. *To our best knowledge, there is no published work that can completely mitigate BPDA, since it is very challenging for defense if attackers can iteratively strengthen the adversarial examples according to the defense. However, we believe our work provides a new angle*

to redesign input-based defense to well balance the accuracy of benign data and defense efficiency with DNN-oriented/defensive quantization. It is a new trial towards developing better input-transformation based defenses.

5.1 Background, Related Work and Motivation

5.1.1 Basics of Adversarial Examples and JPEG

Adversarial examples: ($X^* = X + \delta_X$) are created to fool the DNNs ($Y^* \neq Y$) with imperceptible perturbations: $\arg\min_{\delta_X} \parallel \delta_X \parallel \ s.t.\ F^{(\Theta)}(X + \delta_X) = Y^*$, which can be solved through many crafting algorithms: 1) **FGSM** [39] (fast gradient sign method) is a L_∞ attack and utilizes the gradient of the loss function to determine the direction to modify all the input pixels. It is designed to be fast, rather than optimal; 2) **BIM** [58] (basic iterative gradient sign method) is the iterative version of FGSM by gradually adding small perturbations α (L_∞) until reaching the upper bound ϵ or achieving successful attack; 3) **Deepfool** [76] uses geometrical knowledge to search the minimal perturbations (L_2) by assuming DNN as a linear classifier and each class is separated by a hyper-plane. Such an approach finds the nearest hyper-plane from X and uses geometrical knowledge to calculate the projection distance; 4) **C&W** [16] (Carlini & Wagner method) are a series of L_0, L_2, and L_∞ attacks that achieve 100% attack success rate with much lower distortions comparing with the above-mentioned attacks. In particular, the C&W L_2 attack uses a more effective objective function $f(x) = max\left(max\left\{Z(X)_i \ \ i \neq t\right\} - Z(X)_t, -\kappa\right)$ with logits $Z(X)_i$ and adjustable parameter κ. Further, C&W L_0 and L_∞ attacks are implemented indirectly by iteratively calling their L_2 attack. 5) **BPDA** [8] is the latest adaptive attack by recurrently computing the adversarial gradient after applying defense: $x^* = clip(x + \epsilon \cdot sgn(\nabla_x J_{\theta,Y}(DEF(x))))$, where J represents the function of an DNN

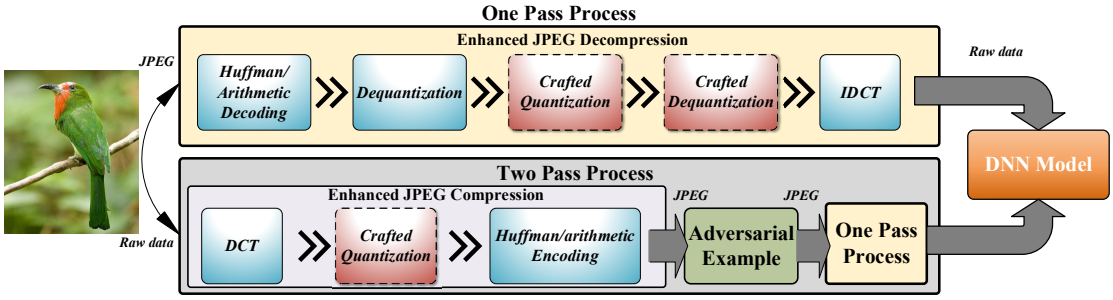

Figure 5.1: Illustration of two different modes of "feature distillation"–one pass and two pass.

model and DEF is the applied defense method in BPDA attack. It is state-of-the-art of attack by assuming adversaries know the defense method.

JPEG: [109] is a popular lossy compression standard for digital images based on the fact that Human-Visual System (HVS) is less sensitive to the high frequency components than low frequency ones [125]. A typical JPEG compression mainly consists of image partitioning, discrete cosine transformation (DCT), quantization, zig-zag reordering and entropy coding, etc. [109]. To compress a raw image, the high (low) frequency DCT coefficients are usually scaled more (less) and then rounded to nearest integers by performing element-wise division based on a predefined 8×8 Quantization Table (Q-Table) [109]. The trade-off between image quality and compression rate is realized by scaling each element in Q-Table via the "Quantization Factor" (QF) [123]. A higher compression rate corresponds to a lower QF. A reverse procedure of above steps can decompress an image.

5.1.2 Related Works

Applying JPEG compression to mitigate adversarial examples has been discussed in prior work. Kurakin et al. [58] test some model-agnostic approaches on adversarial examples and reveal a good potential of JPEG compression for defending adversarial attacks. Dziugaite et al. [35] empirically report JPEG compression can reverse only small adversarial

perturbations, but the reason behind is uncertain. Guo et al. [41] test JPEG compression, image Quilting (piecing together small patches from a database of image patches), total variance minimization (combining pixel dropout with total variation minimization), etc. against the gray-box and black-box adversarial attacks, and report Quilting and TVM show better efficiency than JPEG. Aydemir et al. [9] compare the effects of JPEG compression and JPEG2000, against adversarial perturbations. Though JPEG2000 shows better performance than JPEG, the efficiency is still far from satisfactory. Xu et al. [112] propose an ensemble method, namely “feature squeezing", to defeat the adversarial examples by integrating different types of “squeezers” (i.e. model-agnostic processing). Das et al. [29] propose a JPEG compression based ensemble method, namely “vaccinating", to mitigate adversarial attacks by voting the result based on a variety of compression rates. Prakash et al. [81] develop “pixel deflection" and “adaptive soft-thresholding" approaches by replacing or smoothing adversary perturbations. This method shows good defense efficiency on gray box-setting without evaluating adaptive attacks. Xie. et al. [111] propose two randomization operations–random size and random padding, against adversary examples. *In summary, prior studies empirically test the JPEG compression by directly tuning the compression rate, without digging into the underlying image processing mechanisms. The conclusion is that JPEG suffers from very limited defense efficiency but inevitable DNN accuracy degradation. To overcome those issues, standard JPEG compression should be integrated with the costly ensemble solutions. On the other side, our work directly targets the fundamental entities of JPEG compression/decompression, like defensive and DNN-oriented quantizations, to unleash its defense potentials with almost zero loss of DNN testing accuracy, thus is low-cost.*

5.1.3 Why standard JPEG is not good?

DNN suffers from both low testing accuracy and weak defense efficiency against adversarial examples if we directly employ standard JPEG compression based on human-visual system (HVS). To explore how existing compression can impact DNN's testing accuracy, we trained a MobileNet [48] with high quality JPEG images (QF=100, ImageNet), and tested it with both clear images and FGSM-based adversarial examples at various QFs (i.e., QF=100, 90, 75, 50). As Fig. 2.3 (a) shows, the testing accuracy degrades significantly as the compression rate increases (or QF from 100 to 50), despite the slightly improved defense efficiency (or drop in attack success rate). To achieve the best defense efficiency among our selected four QFs (attack success rate = 0.62 at QF = 50), the accuracy is even reduced by $\sim 8\%$ on benign images than that of the original one (QF=100). Apparently, the HVS-based JPEG compression is not an ideal solution in terms of defense efficiency and accuracy. Fig. 2.3 (b) further shows the means and standard deviations of DCT coefficients of malicious distortions at all 64 frequency bands. Given that malicious perturbations are almost randomly distributed in every frequency band, HVS-based JPEG compression, which distorts more (less) on high (low) frequency components of the input, is unlikely to effectively filter the distortions across the whole spectral domain.

5.2 Our Approach–Feature Distillation

In this section, we first provide a detailed analysis on how to wisely redesign the quantization process in JPEG compression to minimize attack success rate. As this lossy compression will still reduce the classification accuracy (see Fig. 2.3), we then develop the DNN-oriented quantization refine method, to compensate the reduced accuracy of benign images. Based on how/where the derived quantization will be placed in JPEG, our framework supports two modes (see Fig. 5.1): 1) **One pass process** by inserting a new

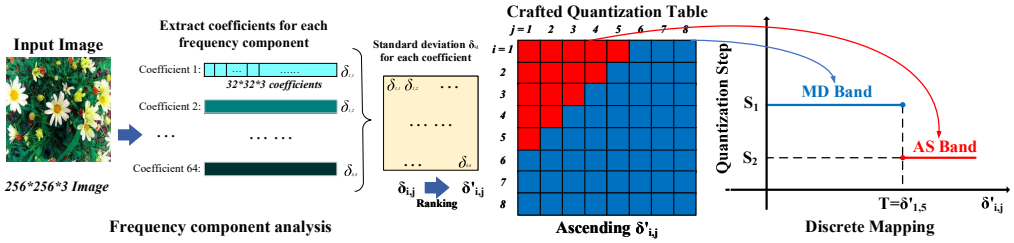

Figure 5.2: An overview of heuristic design flow of DNN-Oriented compression based on crafted quantization.

quantization/de-quantization only in the decompression of standard JPEG; 2) **Two pass process** by also replacing the quantization of compression, followed by one pass process. The two pass method provides an opportunity to directly embed crafted quantization at sensor side to compress raw data to further improve defense efficiency, given that JPEG-based image compression, an integrated component in sensors, is usually a "must-have" step to save data storage/transfer cost in real applications. *Therefore, the one-pass handles incoming images compressed by standard JPEG before sending them to DNNs, while the two-pass targets raw data directly sampled by devices like image sensors.* The target is to address both attack efficiency and test accuracy simultaneously.

5.2.1 Step 1: Defensive Quantization for Enhancing Defense

We propose to use spectral filter by leveraging quantization process in JPEG on DNN inputs (i.e., adversarial examples), in order to mitigate adversarial perturbations.

One pass process. The idea is to directly filter out the malicious perturbations in frequency domain through the quantization process. As Fig. 5.1 shows, the JPEG-formatted input will be decompressed and then feed into the DNNs as the raw data at the beginning. By taking this chance, we insert a new pair of quantization/dequantization processes after the dequantization of standard JPEG decompression to purify the potential adversarial perturbations. Note we omit the first dequantization in the following analysis ideally by

assuming it can almost preserve all frequency features of the input. Assuming for each 8×8 block in the input image X, adversarial distortion δ_X is added to X with intensity ϵ. The DCT transformation–a linear operation, essentially projects the image from spatial domain to spectral domain. Therefore, the original input and adversarial perturbations could be linearly separated as:

$$DCT(X + \delta_X) = DCT(X) + DCT(\delta_X) = C_X \cdot B + C_{\delta_X} \cdot B \tag{5.1}$$

where C_X and C_{δ_X} are the DCT coefficients of X and δ_X, respectively, for the 8×8 image block, and B is the DCT transformation basis. The maximum magnitude of C_{δ_X} can be calculated by the summation of all 64 frequency components and each term is bounded by $cos(\theta) \cdot \epsilon$. Thus we have $-8 \cdot \epsilon < C_{\delta_X} < 8 \cdot \epsilon$.

The DCT coefficients will be quantized again in this decompression process, providing a good opportunity for filtering the perturbations. The quantization is approximated as:

$$Round\left(^{C_X + C_{\delta_X}}/_{QS}\right) \approx Round\left(^{C_X}/_{QS}\right) + Round\left(^{C_{\delta_X}}/_{QS}\right) \tag{5.2}$$

where QS is the defensive quantization step (QS). Ideally, if $QS > |C_{\delta_X}|$, then the perturbation C_{δ_X} can be eliminated. However, this equation may induce undesired rounding error to limit the efficiency of removing malicious perturbations, given that C_{δ_X} is usually much smaller than C_X. We further analyze such a rounding error by decomposing $C_X = \eta + {}^{QS}/_2$, then we have:

$$Round\left(^{C_X + C_{\delta_X}}/_{QS}\right) = Round\left(^{\eta + QS/2 + C_{\delta_X}}/_{QS}\right) \tag{5.3}$$

If ${}^{QS}/_2 + C_{\delta_X} > QS$, this part will be rounded to $\pm 1, \pm 2, \pm 3...$, which will induce a stronger rounding error than the adversarial perturbations, resulting in degraded defense efficiency.

Two pass process. To avoid such rounding error, we further propose two pass method. As Fig. 5.1 shows, the raw data (i.e. sampled by sensors) will be compressed through a

defensive quantization process, rather than the standard JPEG quantization, followed by an entire one pass process.

Assuming such compressed benign inputs are then polluted by adversarial perturbations, adversarial examples will be further processed by considering both compression/decompression procedures as:

$$Round\left((Round\left(\frac{\eta+QS/2}{QS}\right)*QS+C_{\delta_X})/QS\right) = Round(\eta) \tag{5.4}$$

The malicious perturbations can be appropriately filtered without inducing any rounding error if QS satisfies the following equation:

$$Round\left(C_{\delta_X}/QS\right) = 0 \Rightarrow QS > 2\left|C_{\delta_X}\right|,\ C_{\delta_X} \in (-8\epsilon < C_{\delta_X} < 8\epsilon) \tag{5.5}$$

Therefore, we adopt the same QS ($QS > 16 \cdot \epsilon$) to eliminate the perturbations C_{δ_X} in both passes.

5.2.2 Step 2: DNN-Oriented Quantization for Compensating Accuracy Reduction

To recover the testing accuracy (see Section 5.1.3), our next step is to develop a DNN-oriented JPEG compression method by refining the defensive quantization table from step 1. We analyze the difference between human visual system (HVS) and DNN on feature extractions, and then propose a heuristic design flow.

Difference between HVS&DNN on Feature Extractions. Since the feature loss happens in the frequency domain after the DCT process, we first study the problem that which frequency components have the most significant impact on DNN results. Assume x_k

is a single pixel of a raw image X, and x_k can be represented by 8×8 2D DCT transform:

$$x_k = \sum_{i=0}^{7} \sum_{j=0}^{7} c_{(k,i,j)} \cdot b_{(i,j)} \tag{5.6}$$

where $c_{(k,i,j)}$ and $b_{(i,j)}$ are the DCT coefficient and its basis function at 64 different frequencies, respectively. It is well known that the human visual system (HVS) is less sensitive to high frequency components but more sensitive to low frequency ones. The JPEG quantization table is designed based on this fundamental understanding. However, DNNs examine the importance of the frequency information in a quite different way. The gradient of the DNN function F with respect to a basis function $b_{(i,j)}$ is calculated as:

$$\partial F / \partial b_{(i,j)} = \partial F / \partial x_k \times \partial x_k / \partial b_{i,j} = \partial F / \partial x_k \times c_{(k,i,j)} \tag{5.7}$$

Eq. (5.7) implies that the contribution of a frequency component ($b_{i,j}$) to the DNN result will be mainly decided by its associated DCT coefficient ($c_{(k,i,j)}$) and the importance of the pixel ($\partial F / \partial x_k$). Here $c_{(k,i,j)}$ will be distorted by the quantization before training. Ideally a well trained DNN model should respond with different strengths to all the 64 frequency components depending on the $c_{(k,i,j)}$ values. From this observation, large $c_{(k,i,j)}$ should be compressed less (using a small quantization step) in order to ensure a desirable classification accuracy.

In contrast, the default quantization table used in JPEG focuses on compressing more on less sensitive frequency components to HVS. As a result, in order to defend against adversarial attacks, aggressive compression is required, making DNNs easily misclassified if original versions contain important high frequency features. The DNN models trained with original images learn comprehensive features, especially high frequency ones. However, such features are actually lost in more compressed testing images, resulting in considerable misclassification rate (see Fig. 2.3(a)).

Therefore, we propose to compensate the accuracy reduction incurred by defending adversarial examples through a heuristic design flow (see Fig. 5.2): 1) characterize the

importance of each frequency component through frequency analysis on benign images; 2) lower the quantization step of the most sensitive frequency components based on the statistical information for enhancing accuracy.

A: Frequency Component Analysis. For each input image, we first characterize the pre-quantized DCT coefficient distribution at each frequency component. Such a distribution represents the energy contribution of each frequency band [83]. Prior work [83] has proved that the pre-quantized coefficients can be approximated as normal (or Laplace) distribution with zero mean but different standard deviations ($\delta_{i,j}$). A larger $\delta_{i,j}$ means more energy in band (i, j), hence more important features for DNN learning. As Fig. 5.2 shows, each image will be first partitioned into N 8×8 blocks, followed by a block-wise DCT. Then the DCT coefficient distribution at each frequency component will be characterized by sorting all coefficients at the same frequency component across all image blocks. The statistical information, such as the standard deviation $\delta_{i,j}$ of each coefficient, will be calibrated from each individual histogram.

B: Quantization Table Refinement. Once the importance of frequency components is identified based on the standard deviations of DCT coefficients ($\delta_{i,j}$), the next step is to boost the accuracy of legitimate examples $\{acc_l\}$ (refer to the testing accuracy of benign images processed after the defense method). Our analysis in Section 5.1.3 indicates that a proper selection of QS can effectively mitigate the perturbations, whereas larger QS will induce more quantization errors. Therefore, we reduce the quantization errors of the most sensitive frequency components to enhance the testing accuracy by lowering their corresponding quantization steps within the quantization table, but such frequency components should be as few as possible to maintain the defense efficiency. In specific, we first sort the magnitude of $\delta_{i,j}$ in an ascending order as $\delta^{'}_{i,j}$, then set the appropriate quantization step based on $\delta^{'}_{i,j}$. To simplify our design, we introduce a discrete mapping function to derive the quantization step on each frequency band, base on the associated

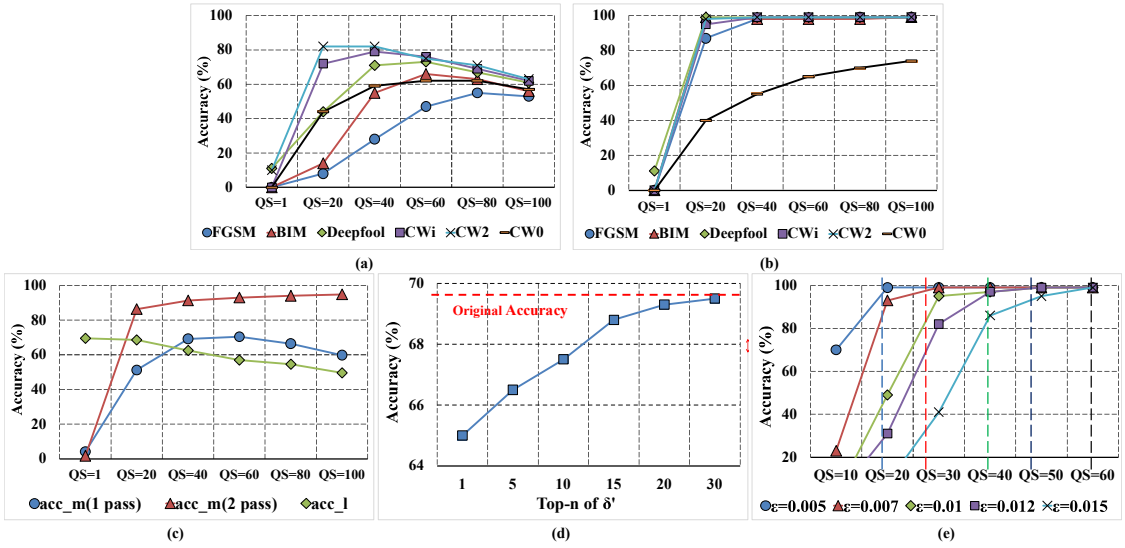

Figure 5.3: Exploration of the defensive quantization step for a 8*8 table: (a) Defense efficiency of one pass method against adversarial examples; (b) Defense efficiency of two pass method against adversarial examples; (c) Average defense efficiency w.r.t. the legitimate image accuracy (FGSM, $\epsilon = 0.008$); (d) Accuracy impacts of ranked frequency components (FGSM, $\epsilon = 0.008$); (e) Accuracy impacts of various quantization steps w.r.t. different perturbation strength (FGSM).

standard deviation $\delta_{i,j}$, i.e., $QS_{i,j} = (\delta_{i,j} \leq T \ ? \ S_1 : S_2)$, where T is the threshold to divide the 64 frequency components. Note that $S_1 > S_2$. The 64 frequency components are divided into two bands (see Fig. 5.2): the red colored Accuracy Sensitive (AS) band with $QS = S_2$, and the blue colored Malicious Defense (MD) band with $QS = S_1$ from Section 5.2.1.

5.3 Evaluation

In this section, we first explore the parameter optimization in our feature distillation under the constraints of high classification accuracy on malicious inputs after applying defense, while preserving the accuracy of legitimate ones given that both types of data can arrive for a realistic DNN testing. Then we comprehensively evaluate feature distillation under following three different settings: 1). **Gray-box:** We assume the adversary has full access

to DNN model, but is unaware of the input transformations applied (defense method unaware) [41, 30]. 2). **White-box:** We consider adversary has full access to the DNN model, as well as the full knowledge of the defense method [8], which is more challenging. 3). **Black-box:** We assume adversary does not know the exact network architecture and weights, instead, can use a substitute model to craft adversarial perturbations that are transferable to the target model [41].

5.3.1 Experimental Setup

Our experiments are conducted on the Tensorflow DNN computing framework [3], running with Intel(R) Xeon(R) 3.5GHz CPU and two parallel GeForce GTX 1080Ti GPUs. Our proposed feature distillation method is implemented on the heavily modified adversarial machine learning library–EvadeML-Zoo [112] for white and gray-box settings and BPDA attack [8] for white box setting. To better illustrate the image compression based mitigation, we choose the large-scaled ImageNet dataset as our benchmark. Four other input-based countermeasures, including default JPEG [35, 58], bit-depth (one of the feature squeezing methods by reducing the bit number of an image pixel) [112] and the recent proposed TV Minimization (TVM) and Image Quilting [41] , are selected as the baselines to compare with our proposed feature distillation.

Methodology. Various types of adversarial example attacks, i.e., FGSM, BIM, Deepfool, CW_2, CW_0, CW_∞ and adaptive attack–BPDA, have been simulated in our experiments for evaluating the defense. We adopt a similar evaluation model from [112]. First, we choose 1000 benign images (one per class) to evaluate the testing accuracy of each DNN model. The seed images, which will be adding adversarial perturbations, are selected from the first 100 correctly predicted examples in the 1000 selected images on each DNN model for all the attack methods. The legitimate examples *classification accuracy* (acc_l)

Table 5.1: The defense efficiency (classification accuracy on adversarial examples) of selected defense methods against different adversarial attacks.

	FGSM	BIM	DeepFool	CW2	CW0	CWi	Average	acc_l	Time (s)
No defense (%)	0	0	11	10	0	0	3.5	69.5	0.11
Bit-depth (5-bit) (%)	2	0	21	68	7	33	21.83	**69.4**	0.04
JPG (90) (%)	5	9	9	68	5	32	21.33	69	0.11
Quilting (%)	48	61	47	50	48	49	50.5	63.5	32.47
TVM (%)	33	42	68	77	49	90	59.8	60	38.89
FD-1P (%)	13	35	63	86	61	78	56	68.5	0.16
FD-2P (%)	**92**	**99**	**99**	**99**	58	**99**	**91**	68.5	0.16

is the testing accuracy of benign images processed by the defense method. The *defense efficiency* is measured by the classification accuracy (acc_m) of 100 polluted images after applying the defense method.

5.3.2 Optimized Quantization Step

Defending against adversarial examples. Fig. 5.3 (a) and (b) illustrate the impact of the quantization steps of the 8*8 table under various adversarial attacks with our one-pass and two-pass defense approaches applied, respectively. Apparently, both processes demonstrate that the defense efficiency can be steadily improved as the QS grows, however, it will be saturated (even decreased) if QS becomes too large for the two pass (one pass) process. Compared with the one pass process, the two pass process always delivers much better defense efficiency against most of the adversarial attacks (except the CW_0), due to the elimination of the rounding error. The reason is because CW_0 attacks attempt to use a minimum number of pixel(s) with maximum perturbations to fool the DNN models, therefore the perturbations of each single pixel will translate into larger magnitudes than the other attacks in the frequency domain. This leads to a much higher QS for completely removing the associated perturbations, as Fig. 5.3 (a) and (b) show.

Evaluating testing accuracy. Fig. 5.3 (c) shows the testing accuracy changes w.r.t. QSs for both malicious examples (acc_m) and legitimate examples (acc_l). The acc_m (1

Table 5.2: The defense efficiency (accuracy on adversarial examples–acc_m and accuracy on legitimate images–acc_l on ImageNet, against adaptive adversarial attack–BPDA.

	None	Bit-depth	Quilting	TVM	JPEG(75)	JPEG(20)	JPEG(10)	FD(1x)	FD(2x)	FD(3x)
acc_m (%)	0	0	0	0	0	34	45	10	42	60
acc_l (%)	78	77	72	68	74	68	61	77	76	74

pass) and acc_m (2 pass) represent the average accuracy of various adversarial examples by applying our one pass and two pass process, respectively. As Fig. 5.3 (c) shows, acc_m (1 pass) and acc_l demonstrate an opposite trend as QS grows, but they have a cross-over zone between QS=20 and QS=40. The adversarial perturbation dominates the accuracy reduction before the cross-over point (small QS), however, after that, both acc_m (1 pass) and acc_l will decrease due to the enlarged QS. On the other hand, acc_m (2 pass) increases consistently as QS increases because of additional defensive quantization in compression stage. *Therefore, we set S_2=20 and S_1=30 for the top-n largest $\delta'_{i,j}$ (AS Band) and the others (MD Band), respectively, to better balance the acc_m and acc_l, according to our flow in Fig. 5.1. Fig. 5.3 (d) validates that such a configuration of (S_1, S_2) at $n = 15$ minimizes the degradation of acc_l ($\leq 1\%$).*

Theoretical validation of defensive QS. Fig. 5.3 (e) further compare our analytic results (see Eq. 5.5) with experimental results for selecting QS. We use FGSM attack with 5 different perturbation strengths (i.e. $\epsilon = 0.005, 0.007, 0.01, 0.012, 0.015$) as an example. The corresponding analytic QS values based on Eq. 5.5 should be: 20.5, 28.7, 41, 49.2 and 61.44, respectively (dash lines in Fig. 5.3 (e)). As expected, those analytical values are in excellent agreement with the experimental results when the defense efficiencies reach 100%.

5.3.3 Enhanced Robustness Against AE

Based on our explorations on parameters optimization in section 5.3.2, we adopt $S_1 = 30, S_2 = 20, n = 15$ to evaluate the overall defense efficiency. Note although we focus on

defense efficiency, the images compressed by our method still provide acceptable visual quality (detailed results are summarized in the supplemental material).

Gray Box Mitigation

Table. 5.1 compares the defense efficiency of two proposed methods (i.e., 1-pass feature distillation **FD-1P** and 2-pass version **FD-2P**) with five baselines–no defense, JPEG, Bit-depth, Quilting and TVM against 6 selected adversarial examples for MobileNet. *Note that JPEG (90% quality) and Bit-depth (5-bits) are conducted under the premise of* $\leq 1\%$ *legitimate classification accuracy reduction. However, the other two methods Quilting and TVM, cannot satisfy this constraint, so we compare our approach with those two methods on both* acc_l *and* acc_m*.*

Comparison with bit-depth and JPEG. We first limit our comparisons to the defense with $\leq 1\%$ reduction of acc_l under no defense. In this case, Quilting and TVM are not included and will be compared separately.

Overall, FD-2P shows much better performance than that of FD-1P (56% v.s. 91% on average). Compared with no defense baseline, our FD-2P improves the average accuracy on adversarial examples from $\sim 3.5\%$ to $\sim 91\%$, which demonstrates the best mitigation efficiency among all methods. Moreover, FD-2P can significantly outperform two other defensive baselines among all selected adversarial examples, i.e. improved by $\sim 69\%$ (or $\sim 73\%$) on average than the bit-dept (5-bit) or JPEG on both DNN models.

Particularly, for L_∞ attacks like FGSM, BIM and CWi, existing model-agnostic methods show very limited efficiency. Similarly, our one pass method FD-1P shows marginal improvement when compared with the existing approaches. However, our two pass method FD-2P can almost completely remove this type of L_∞ perturbations and deliver the best defense efficiency. Besides, for the L_2 attacks, especially CW_2, existing defense methods show good defense efficiency ($\sim 68\%$). Again FD-2P can rectify this kind of adversarial

examples with almost 100%. Compared with L_∞ and L_2, the improvement of L_0 attacks (CW_0) is less significant, however, FD-2P still achieves more than 50% defense efficiency improvement comparing with bit-depth and JPEG. That is because, JPEG (90% quality) uses small quantization steps (or large QFs) to maintain the quality of legitimate images for desirable accuracy, however, is also resulting in a low defense efficiency. Bit-depth roughly quantizes all image pixels uniformly, while our method distills the features in a more fine-grained manner by maximizing the loss of adversarial perturbations and minimizing the distortions of benign features.

Comparison with Quilting/TVM. We also compare our solutions with Quilting and TVM in three aspects: acc_m, acc_l, and processing-time-per-image. Our average defense efficiency is much higher than the other two, i.e. 56%/91% (FD-1P/FD-2P) v.s. 50.5% (quilting), 59.8% (TVM). We also achieve the best testing accuracy (acc_l), that is 68.5% (FD-1P/2P) v.s. 63.5% (quilting), 60% (TVM). Moreover, we improve the processing-time-per-image (i.e., 0.15s on FD-1P) by $\sim 216\times$ ($\sim 259\times$) compared with Quilting (32.4s) and TVM (38.8s), or 0.15s (FD-1P) v.s. 32.4s (quilting) and 38.8s (TVM), as Table 5.1 shows.

In general, our proposed feature distillation is particularly effective to mitigate stronger attacks (i.e. CW attacks with least perturbations but $\sim 100\%$ attack success rate) crafted from complex datasets like ImageNet. Our solution demonstrates great potentials to safeguard the DNNs against adversarial attacks in practical applications, given that it is likely the attackers prefer to generate stronger adversarial examples with minimum adversarial perturbations from realistic large-scale dataset so as to evade any possible defense methods.

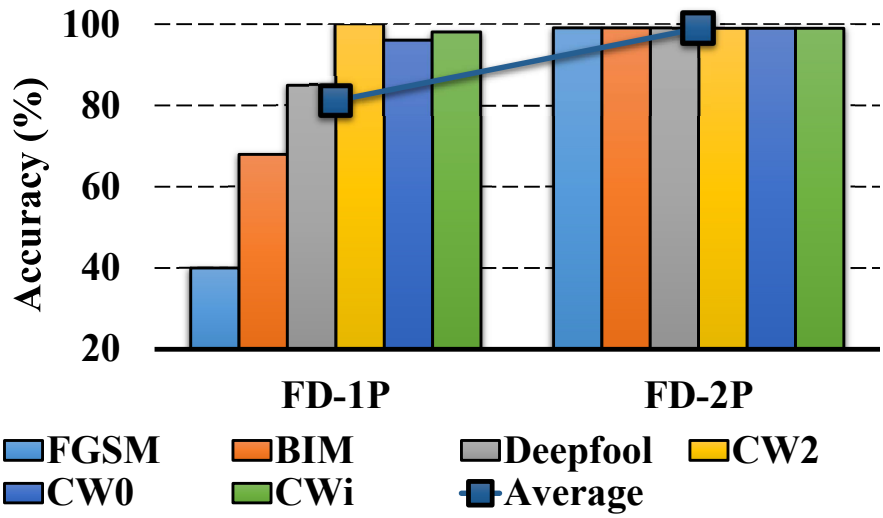

Figure 5.4: Defense efficiency of black-box setting for different attack and defense mechanisms on ImageNet.

White Box Mitigation

In this section, we evaluate our method against recent BPDA attack, of which adversary knows the defense method and iteratively generates adversarial examples according to the defense. We implement our defense–Feature Distillation (FD-1P) in the released BPDA attack [8] code at GitHub, using the same "Inception v3" model and 100 iterations for BPDA. The accuracy of benign examples (adversarial examples) after defense-acc_l (acc_m), for different methods are reported in Table 5.2.

First, Bit-depth, quilting and TVM does not offer any defense against BPDA, as expected. **Second**, JPEG can slightly mitigate BPDA by degrading image quality, i.e. quality factor from 75 to 10, defense efficiency (acc_m) is improved from 0 to 45%. This is consistent with the recent result [90]. However, acc_l drops by 17% compared to baseline (61% v.s. 78%), which is unacceptable. This reason is because in order to eliminate a large perturbation of BPDA attack in the lowest frequency component in JPEG, a significant large quantization factor (QF) will be needed. As a result, large quantization errors will occur in high frequency components, thereby significantly hurting acc_l. **Third**, On the

other hand, our solution can provide the best defense efficiency against BPDA with least acc_l reduction among all solutions, i.e. from FD (1X) to FD (3X), acc_m is improved from 10% to 60%, with merely 1%-3% acc_l reduction compared to original 78%. FD-1$\times$, 2$\times$, 3$\times$ represent the quantization step (QS) of FD adopted in Table. 5.1 (reference), 2 times and 3 times of the referred QS, respectively.

Black Box Mitigation

We follow the work [41] for black-box analysis: DNN model used for testing is trained on transformed dataset (Feature Distillation), while attackers generate adversarial examples from the model trained on the original dataset. The crafted examples have high transferability between the two models for fair black-box analysis. We adopt "MobileNet" and the results of our methods are shown in Fig. 5.4.

The average defense efficiency is improved from 56%/91% (Table 5.1) to 81%/99% (black-box) for our FD-1P/FD-2P method, respectively. These results indicate that our method defends against black-box setting efficiently. This is also consistent with the following conclusion based on [8, 41]: Black box setting shows weak attack efficiency against the input-transformation based defenses.

Comparison of Visual Quality–Quantitative

As Table. 5.3 shows, all these three images compressed by our method (FD(1x)) can achieve reasonable PSNR and SSIM, e.g. close to that of $QF = 75$ for JPEG, which is still acceptable for most visual systems. Similarly, the PSNR and SSIM of our FD(2x) and FD(3X) are comparable with JPEG method at $QF = 50$ and $QF = 20$, respectively.

Table 5.3: The comparision of PSNR/SSIM between “feature distillation" (FD) and JPEG.

	Img. 1		Img. 2		Img. 3	
	PSNR	**SSIM**	**PSNR**	**SSIM**	**PSNR**	**SSIM**
JEPG(QF=75)	33.63	0.94	28.64	0.94	35.64	0.97
JEPG(QF=50)	31.41	0.92	26.05	0.89	33.75	0.96
JEPG(QF=20)	28.81	0.87	23.56	0.82	31.03	0.94
FD(1x)	33.05	0.93	29.12	0.94	34.86	0.97
FD(2x)	30.03	0.89	26.29	0.89	32.53	0.95
FD(3x)	28.44	0.86	24.15	0.84	31.11	0.94

5.4 Conclusion

As the robustness of DNN is significantly challenged by a variety of adversarial attacks, existing studies investigate the standard JPEG compression as a defense method, however, it is far from satisfactory in terms of both defense efficiency and testing accuracy. In this work, we propose the DNN-favorable feature distillation method by re-architecting the JPEG compression framework. Compared with existing model-agnostic defense approaches, our “feature distillation" can simultaneously reduce the adversarial attack success rate and maximize the testing accuracy on legitimate examples. Experimental results show that our method can improve the defense efficiency from $\sim 20\%$ to $\sim 90\%$ over most recent model-agnostic approaches with only marginal accuracy degradation ($\leq 1\%$), while significantly improving the processing time per image ($\sim 260\times$ speedup). Our method also demonstrates the best defense efficiency against latest adaptive attack–BPDA ($\sim 60\%$) with least accuracy drop ($\sim 1\%$) when compared with other input-transformation based defenses.

Figure 5.5: Visual results produced by default JPEG compression.

Figure 5.6: Visual results produced by “feature distillation" method.

CHAPTER 6

CONCLUSIONS

The ever-increasing data transfer and storage overhead significantly challenge the energy efficiency, latency and performance of large-scale DNNs. In this dissertation we find that human visual system (HVS) and Deep Learning have very different views on the importance of image features, challenging HVS-oriented compression for communication overhead and robustness against adversarial attacks.

First of all, we propose a DNN oriented image compression framework, namely "DeepN-JPEG", to ease the storage and data communication overhead. Instead of the Human Vision System inspired JPEG compression, our solution effectively reduces the quantization error based on the frequency component analysis and rectified quantization table, and further increases the compressing rate without any accuracy degradation. Our experimental results show that "DeepN-JPEG" achieves $\sim 3.5\times$ compression rate improvement, and consumes only 30% power of the conventional JPEG without classification accuracy degradation, representing a promising solution for data storage and communication for deep learning.

Due to the high computation complexity of DNNs and the increasingly large volume of medical images, cloud based medical image segmentation has become popular recently. Medical image transmission from local to clouds is the bottleneck for such a service, as it is much more time-consuming than neural network processing on clouds. We then extend our work to medical 2D/3D image tasks, we develop a low cost machine vision guided 3D image compression framework dedicated to DNN-based image segmentation by taking advantage of such differences between human vision and DNN. Extensive experiments on widely adopted segmentation DNNs with HVSMR 2016 challenge dataset show that our method significantly beats existing 3D JPEG-2000 in terms of segmentation accuracy and compression rate.

We also desgin a generative segmentation architecture to further compress biomedical images, which consists of a compressive auto-encoder, a segmentation network and a discriminator network. We propose to leverage the auto-encoder and different loss function designs to enhance the cloud-based segmentation performance and efficiency by synthetically considering segmentation accuracy and compression rate. Experimental results show that our design not only significantly improves compression rate, but also increases the segmentation accuracy, outperforming existing solutions by offering better efficiency on cloud-based image segmentation.

In addition, as the robustness of DNN is significantly challenged by a variety of adversarial attacks, existing studies investigate the standard JPEG compression as a defense method, however, it is far from satisfactory in terms of both defense efficiency and testing accuracy. We propose the DNN-favorable feature distillation method by re-architecting the JPEG compression framework. Compared with existing model-agnostic defense approaches, our proposed compression framework can simultaneously reduce the adversarial attack success rate and maximize the testing accuracy on legitimate examples. To summarize, we believe that “Machine Vision" concept advocated by this dissertation, rather than “Human Vision", should be a new angle for developing innovative designs that can provide low-latency, efficient and robust deep learning services.

BIBLIOGRAPHY

[1] Microsoft Azure Computer Vision API Version 1.0, `https://docs.microsoft.com/en-us/azure/cognitive-services/computer-vision/home`,.

[2] Openjpeg jpeg 2000 compression library. http://www.openjpeg.org/.

[3] Martín Abadi, Paul Barham, et al. Tensorflow: A system for large-scale machine learning. In *OSDI*, volume 16, pages 265–283, 2016.

[4] Agustsson and etc. Generative adversarial networks for extreme learned image compression. *arXiv:1804.02958*, 2018.

[5] Naveed Akhtar and Ajmal Mian. Threat of adversarial attacks on deep learning in computer vision: A survey. *arXiv preprint arXiv:1801.00553*, 2018.

[6] Daniel Andor, Chris Alberti, David Weiss, Aliaksei Severyn, Alessandro Presta, Kuzman Ganchev, Slav Petrov, and Michael Collins. Globally normalized transition-based neural networks. *arXiv preprint arXiv:1603.06042*, 2016.

[7] Marc Antonini, Michel Barlaud, Pierre Mathieu, and Ingrid Daubechies. Image coding using wavelet transform. *IEEE Transactions on image processing*, 1(2):205–220, 1992.

[8] Anish Athalye, Nicholas Carlini, and David Wagner. Obfuscated gradients give a false sense of security: Circumventing defenses to adversarial examples. *arXiv preprint arXiv:1802.00420*, 2018.

[9] Ayse Elvan Aydemir, Alptekin Temizel, and Tugba Taskaya Temizel. The effects of jpeg and jpeg2000 compression on attacks using adversarial examples. *arXiv preprint arXiv:1803.10418*, 2018.

[10] Arjun Nitin Bhagoji, Daniel Cullina, and Prateek Mittal. Dimensionality reduction as a defense against evasion attacks on machine learning classifiers. *arXiv preprint arXiv:1704.02654*, 2017.

[11] Christopher M Bishop. *Pattern recognition and machine learning*. springer, 2006.

[12] Mariusz Bojarski, Davide Del Testa, Daniel Dworakowski, Bernhard Firner, Beat Flepp, Prasoon Goyal, Lawrence D Jackel, Mathew Monfort, Urs Muller, Ji-

akai Zhang, et al. End to end learning for self-driving cars. *arXiv preprint arXiv:1604.07316*, 2016.

[13] Martin Boliek. Information technology jpeg 2000 image coding system: Extensions for three-dimensional data. *ISO/IEC 15444-10, ITU-T Rec. T.809*, 2002.

[14] Martin Boliek. Jpeg 2000 image coding system: Core coding system. *ISO/IEC*, 2002.

[15] Tim Bruylants, Adrian Munteanu, and Peter Schelkens. Wavelet based volumetric medical image compression. *Signal processing: Image communication*, 31:112–133, 2015.

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

[17] Varun Chandola, Arindam Banerjee, and Vipin Kumar. Anomaly detection: A survey. *ACM computing surveys (CSUR)*, 41(3):15, 2009.

[18] Jianshu Chao, Hu Chen, and Eckehard Steinbach. On the design of a novel jpeg quantization table for improved feature detection performance. In *Image Processing (ICIP), 2013 20th IEEE International Conference on*, pages 1675–1679. IEEE, 2013.

[19] Hao Chen, Qi Dou, Lequan Yu, Jing Qin, and Pheng-Ann Heng. Voxresnet: Deep voxelwise residual networks for brain segmentation from 3d mr images. *NeuroImage*, 2017.

[20] Hao Chen, Xiaojuan Qi, Jie-Zhi Cheng, Pheng-Ann Heng, et al. Deep contextual networks for neuronal structure segmentation. In *AAAI*, pages 1167–1173, 2016.

[21] Hao Chen, Xiaojuan Qi, Lequan Yu, and Pheng-Ann Heng. Dcan: Deep contour-aware networks for accurate gland segmentation. In *CVPR*, pages 2487–2496, 2016.

[22] Jianxu Chen, Lin Yang, Yizhe Zhang, Mark Alber, and Danny Z Chen. Combining fully convolutional and recurrent neural networks for 3d biomedical image segmentation. In *Advances in Neural Information Processing Systems*, pages 3036–3044, 2016.

[23] Chiou and etc. A complexity analysis of the jpeg image compression algorithm. In *2017 9th Computer Science and Electronic Engineering (CEEC)*, pages 65–70. IEEE, 2017.

[24] Özgün Çiçek, Ahmed Abdulkadir, Soeren S Lienkamp, Thomas Brox, and Olaf Ronneberger. 3d u-net: learning dense volumetric segmentation from sparse annotation. In *International Conference on Medical Image Computing and Computer-Assisted Intervention*, pages 424–432. Springer, 2016.

[25] Codella and etc. Skin lesion analysis toward melanoma detection: A challenge at the 2017 international symposium on biomedical imaging (isbi), hosted by the international skin imaging collaboration (isic). In *2018 IEEE 15th International Symposium on Biomedical Imaging (ISBI 2018)*, pages 168–172. IEEE, 2018.

[26] R. Collobert, K. Kavukcuoglu, and C. Farabet. Torch7: A matlab-like environment for machine learning. In *BigLearn, NIPS Workshop*, 2011.

[27] Werner Coomans, Rodrigo B Moraes, Koen Hooghe, Alex Duque, Joe Galaro, Michael Timmers, Adriaan J van Wijngaarden, Mamoun Guenach, and Jochen Maes. Xg-fast: the 5th generation broadband. *IEEE Communications Magazine*, 53(12):83–88, 2015.

[28] George E Dahl, Dong Yu, Li Deng, and Alex Acero. Context-dependent pre-trained deep neural networks for large-vocabulary speech recognition. *IEEE Transactions on audio, speech, and language processing*, 20(1):30–42, 2012.

[29] Nilaksh Das, Madhuri Shanbhogue, Shang-Tse Chen, Fred Hohman, Li Chen, Michael E Kounavis, and Duen Horng Chau. Keeping the bad guys out: Protecting and vaccinating deep learning with jpeg compression. *arXiv preprint arXiv:1705.02900*, 2017.

[30] Nilaksh Das, Madhuri Shanbhogue, Shang-Tse Chen, Fred Hohman, Siwei Li, Li Chen, Michael E Kounavis, and Duen Horng Chau. Shield: Fast, practical defense and vaccination for deep learning using jpeg compression. *arXiv preprint arXiv:1802.06816*, 2018.

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

[32] Ivo D Dinov. Volume and value of big healthcare data. *Journal of medical statistics and informatics*, 4, 2016.

[33] Qi Dou, Hao Chen, Yueming Jin, Lequan Yu, Jing Qin, and Pheng-Ann Heng. 3d deeply supervised network for automatic liver segmentation from ct volumes. In *International Conference on Medical Image Computing and Computer-Assisted Intervention*, pages 149–157. Springer, 2016.

[34] Ling-Yu Duan, Xiangkai Liu, Jie Chen, Tiejun Huang, and Wen Gao. Optimizing jpeg quantization table for low bit rate mobile visual search. In *Visual Communications and Image Processing (VCIP), 2012 IEEE*, pages 1–6. IEEE, 2012.

[35] Gintare Karolina Dziugaite, Zoubin Ghahramani, and Daniel M Roy. A study of the effect of jpg compression on adversarial images. *arXiv preprint arXiv:1608.00853*, 2016.

[36] Jim Gerrity. Health networks - delivering the future of healthcare, 2014. `https://www.buildingbetterhealthcare.co.uk/technical/article_page/Comment_Health_networks__delivering_the_future_of_healthcare/94931`.

[37] Alessandro Giusti, Jérôme Guzzi, Dan C Cireşan, Fang-Lin He, Juan P Rodríguez, Flavio Fontana, Matthias Faessler, Christian Forster, Jürgen Schmidhuber, Gianni Di Caro, et al. A machine learning approach to visual perception of forest trails for mobile robots. *IEEE Robotics and Automation Letters*, 1(2):661–667, 2016.

[38] Ian Goodfellow, Jonathon Shlens, and Christian Szegedy. Explaining and harnessing adversarial examples. 2015.

[39] Ian J Goodfellow, Jonathon Shlens, and Christian Szegedy. Explaining and harnessing adversarial examples. *arXiv preprint arXiv:1412.6572*, 2014.

[40] Alex Graves, Abdel-rahman Mohamed, and Geoffrey Hinton. Speech recognition with deep recurrent neural networks. In *Acoustics, speech and signal processing (icassp), 2013 ieee international conference on*, pages 6645–6649. IEEE, 2013.

[41] Chuan Guo, Mayank Rana, Moustapha Cissé, and Laurens van der Maaten. Countering adversarial images using input transformations. *arXiv preprint arXiv:1711.00117*, 2017.

[42] Song Han, Xingyu Liu, Huizi Mao, Jing Pu, Ardavan Pedram, Mark A Horowitz, and William J Dally. Eie: efficient inference engine on compressed deep neural network. In *Proceedings of the 43rd International Symposium on Computer Architecture*, pages 243–254. IEEE Press, 2016.

[43] He and etc. Convolutional neural networks at constrained time cost. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 5353–5360, 2015.

[44] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

[45] Geoffrey Hinton, Li Deng, Dong Yu, George E Dahl, Abdel-rahman Mohamed, Navdeep Jaitly, Andrew Senior, Vincent Vanhoucke, Patrick Nguyen, Tara N Sainath, et al. Deep neural networks for acoustic modeling in speech recognition: The shared views of four research groups. *IEEE Signal Processing Magazine*, 29(6):82–97, 2012.

[46] Geoffrey E Hinton and Ruslan R Salakhutdinov. Reducing the dimensionality of data with neural networks. *science*, 313(5786):504–507, 2006.

[47] Max Hopkins, Michael Mitzenmacher, and Sebastian Wagner-Carena. Simulated annealing for jpeg quantization. *arXiv preprint arXiv:1709.00649*, 2017.

[48] Andrew G Howard, Menglong Zhu, Bo Chen, Dmitry Kalenichenko, Weijun Wang, Tobias Weyand, Marco Andreetto, and Hartwig Adam. Mobilenets: Efficient convolutional neural networks for mobile vision applications. *arXiv preprint arXiv:1704.04861*, 2017.

[49] IJG. Independent jpeg group.

[50] Isensee and etc. nnu-net: Breaking the spell on successful medical image segmentation. *arXiv preprint arXiv:1904.08128*, 2019.

[51] ISO ITU-T and IEC JTC. Generic coding of moving pictures and associated audio information-part 2: video, 1995.

[52] Yangqing Jia, Evan Shelhamer, Jeff Donahue, Sergey Karayev, Jonathan Long, Ross Girshick, Sergio Guadarrama, and Trevor Darrell. Caffe: Convolutional architecture for fast feature embedding. *arXiv preprint arXiv:1408.5093*, 2014.

[53] Weiwen Jiang, Xinyi Zhang, Edwin H-M Sha, Lei Yang, Qingfeng Zhuge, Yiyu Shi, and Jingtong Hu. Accuracy vs. efficiency: Achieving both through fpga-implementation aware neural architecture search. *arXiv preprint arXiv:1901.11211*, 2019.

[54] Weiwen Jiang, Xinyi Zhang, Edwin H-M Sha, Qingfeng Zhuge, Lei Yang, Yiyu Shi, and Jingtong Hu. Xfer: A novel design to achieve super-linear performance on multiple fpgas for real-time ai. In *Proceedings of the 2019 ACM/SIGDA International Symposium on Field-Programmable Gate Arrays*, pages 305–305. ACM, 2019.

[55] Yiping Kang, Johann Hauswald, Cao Gao, Austin Rovinski, Trevor Mudge, Jason Mars, and Lingjia Tang. Neurosurgeon: Collaborative intelligence between the cloud and mobile edge. In *Proceedings of the Twenty-Second International Conference on Architectural Support for Programming Languages and Operating Systems*, pages 615–629. ACM, 2017.

[56] Blossom Kaur, Amandeep Kaur, and Jasdeep Singh. Steganographic approach for hiding image in dct domain. *International Journal of Advances in Engineering & Technology*, 1(3):72, 2011.

[57] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*, pages 1097–1105, 2012.

[58] Alexey Kurakin, Ian Goodfellow, and Samy Bengio. Adversarial examples in the physical world. *arXiv preprint arXiv:1607.02533*, 2016.

[59] Lakhani and etc. Machine learning in radiology: applications beyond image interpretation. *Journal of the American College of Radiology*, 15(2):350–359, 2018.

[60] Jia Li and Robert M Gray. Text and picture segmentation by the distribution analysis of wavelet coefficients. In *Image Processing, 1998. ICIP 98. Proceedings. 1998 International Conference on*, pages 790–794. IEEE, 1998.

[61] Fangzhou Liao, Ming Liang, Yinpeng Dong, Tianyu Pang, Jun Zhu, and Xiaolin Hu. Defense against adversarial attacks using high-level representation guided denoiser. *arXiv preprint arXiv:1712.02976*, 2017.

[62] Qi Liu, Tao Liu, Zihao Liu, Yanzhi Wang, Yier Jin, and Wujie Wen. Security analysis and enhancement of model compressed deep learning systems under adversarial attacks. In *Proceedings of the 23rd Asia and South Pacific Design Automation Conference*, ASPDAC '18, pages 721–726, Piscataway, NJ, USA, 2018. IEEE Press.

[63] Tao Liu, Lei Jiang, Yier Jin, Gang Quan, and Wujie Wen. Pt-spike: A precise-time-dependent single spike neuromorphic architecture with efficient supervised learning.

In *Design Automation Conference (ASP-DAC), 2018 23rd Asia and South Pacific*, pages 568–573. IEEE, 2018.

[64] Tao Liu, Zihao Liu, Fuhong Lin, Yier Jin, Gang Quan, and Wujie Wen. Mt-spike: A multilayer time-based spiking neuromorphic architecture with temporal error backpropagation. *2017 IEEE/ACM International Conference on Computer-Aided Design (ICCAD)*, pages 450–457, 2017.

[65] Zihao Liu, Wujie Wen, Lei Jiang, Yier Jin, and Gang Quan. A statistical stt-ram retention model for fast memory subsystem designs. In *Design Automation Conference (ASP-DAC), 2017 22nd Asia and South Pacific*, pages 720–725. IEEE, 2017.

[66] Zihao Liu, Xiaowei Xu, Tao Liu, Qi Liu, Yanzhi Wang, Yiyu Shi, Wujie Wen, Meiping Huang, Haiyun Yuan, and Jian Zhuang. Machine vision guided 3d medical image compression for efficienttransmission and accurate segmentation in the clouds. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, accepted, 2019.

[67] Long and etc. Fully convolutional networks for semantic segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3431–3440, 2015.

[68] Luc and etc. Semantic segmentation using adversarial networks. *arXiv preprint arXiv:1611.08408*, 2016.

[69] Luís FR Lucas, Nuno MM Rodrigues, Luis A da Silva Cruz, and Sérgio MM de Faria. Lossless compression of medical images using 3-d predictors. *IEEE transactions on medical imaging*, 36(11):2250–2260, 2017.

[70] Yiran Ma and Zhensheng Jia. Evolution and trends of broadband access technologies and fiber-wireless systems. In *Fiber-Wireless Convergence in Next-Generation Communication Networks*, pages 43–75. Springer, 2017.

[71] Mbarek Marwan, Ali Kartit, and Hassan Ouahmane. Using cloud solution for medical image processing: Issues and implementation efforts. In *Cloud Computing Technologies and Applications (CloudTech), 2017 3rd International Conference of*, pages 1–7. IEEE, 2017.

[72] Jirı Matela et al. Gpu-based dwt acceleration for jpeg2000. In *Annual Doctoral Workshop on Mathematical and Engineering Methods in Computer Science*, pages 136–143, 2009.

[73] Fausto Milletari, Nassir Navab, and Seyed-Ahmad Ahmadi. V-net: Fully convolutional neural networks for volumetric medical image segmentation. In *3D Vision (3DV), 2016 Fourth International Conference on*, pages 565–571. IEEE, 2016.

[74] Minnen and etc. Joint autoregressive and hierarchical priors for learned image compression. In *Advances in Neural Information Processing Systems*, pages 10771–10780, 2018.

[75] Rani Molla. Fixed broadband speeds are getting faster — what's fastest in your city?, 2017.

[76] Seyed Mohsen Moosavi Dezfooli, Alhussein Fawzi, and Pascal Frossard. Deepfool: a simple and accurate method to fool deep neural networks. In *Proceedings of 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, number EPFL-CONF-218057, 2016.

[77] Eshed Ohn-Bar and Mohan Manubhai Trivedi. Looking at humans in the age of self-driving and highly automated vehicles. *IEEE Transactions on Intelligent Vehicles*, 1(1):90–104, 2016.

[78] Danielle F Pace, Adrian V Dalca, Tal Geva, Andrew J Powell, Mehdi H Moghari, and Polina Golland. Interactive whole-heart segmentation in congenital heart disease. In *International Conference on Medical Image Computing and Computer-Assisted Intervention*, pages 80–88. Springer, 2015.

[79] Paszke and etc. Automatic differentiation in pytorch. 2017.

[80] Barbara Penna, Tammam Tillo, Enrico Magli, and Gabriella Olmo. Progressive 3-d coding of hyperspectral images based on jpeg 2000. *IEEE Geoscience and remote sensing letters*, 3(1):125–129, 2006.

[81] Aaditya Prakash, Nick Moran, Solomon Garber, Antonella DiLillo, and James Storer. Deflecting adversarial attacks with pixel deflection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 8571–8580, 2018.

[82] Viresh Ratnakar and Miron Livny. An efficient algorithm for optimizing dct quantization. *IEEE Transactions on Image Processing*, 9(2):267–270, 2000.

[83] R Reininger and J Gibson. Distributions of the two-dimensional dct coefficients for images. *IEEE Transactions on Communications*, 31(6):835–839, 1983.

[84] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *MICCAI*, pages 234–241. Springer, 2015.

[85] Pouya Samangouei, Maya Kabkab, and Rama Chellappa. Defense-gan: Protecting classifiers against adversarial attacks using generative models. *arXiv preprint arXiv:1805.06605*, 2018.

[86] Victor Sanchez, Rafeef Abugharbieh, and Panos Nasiopoulos. Symmetry-based scalable lossless compression of 3d medical image data. *IEEE Transactions on Medical Imaging*, 28(7):1062–1072, 2009.

[87] Victor Sanchez, Rafeef Abugharbieh, and Panos Nasiopoulos. 3-d scalable medical image compression with optimized volume of interest coding. *IEEE Transactions on Medical Imaging*, 29(10):1808–1820, 2010.

[88] João M Santos, André FR Guarda, Nuno MM Rodrigues, and Sérgio MM Faria. Contributions to lossless coding of medical images using minimum rate predictors. In *Image Processing (ICIP), 2015 IEEE International Conference on*, pages 2935–2939. IEEE, 2015.

[89] Jay W Schwartz and Richard C Barker. Bit-plane encoding: a technique for source encoding. *IEEE Transactions on Aerospace and Electronic Systems*, (4):385–392, 1966.

[90] Uri Shaham, James Garritano, Yutaro Yamada, Ethan Weinberger, Alex Cloninger, Xiuyuan Cheng, Kelly Stanton, and Yuval Kluger. Defending against adversarial images using basis functions transformations. *arXiv preprint arXiv:1803.10840*, 2018.

[91] Mark J Shensa. The discrete wavelet transform: wedding the a trous and mallat algorithms. *IEEE Transactions on signal processing*, 40(10):2464–2482, 1992.

[92] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014.

[93] Athanassios Skodras, Charilaos Christopoulos, and Touradj Ebrahimi. The jpeg 2000 still image compression standard. *IEEE Signal processing magazine*, 18(5):36–58, 2001.

[94] Nikolai Smolyanskiy, Alexey Kamenev, Jeffrey Smith, and Stan Birchfield. Toward low-flying autonomous mav trail navigation using deep neural networks for envi-

ronmental awareness. In *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 4241–4247. IEEE, 2017.

[95] Yang Song, Taesup Kim, Sebastian Nowozin, Stefano Ermon, and Nate Kushman. Pixeldefend: Leveraging generative models to understand and defend against adversarial examples. *arXiv preprint arXiv:1710.10766*, 2017.

[96] Stanislava Soro and Wendi Heinzelman. A survey of visual sensor networks. *Advances in multimedia*, 2009, 2009.

[97] Matthew C Stamm and KJ Ray Liu. Anti-forensics of digital image compression. *IEEE Transactions on Information Forensics and Security*, 6(3):1050–1065, 2011.

[98] Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich. Going deeper with convolutions. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1–9, 2015.

[99] Christian Szegedy, Wojciech Zaremba, Ilya Sutskever, Joan Bruna, Dumitru Erhan, Ian Goodfellow, and Rob Fergus. Intriguing properties of neural networks. *arXiv preprint arXiv:1312.6199*, 2013.

[100] David Taubman. High performance scalable image compression with ebcot. *IEEE Transactions on image processing*, 9(7):1158–1170, 2000.

[101] ITU Telecom et al. Advanced video coding for generic audiovisual services. *ITU-T Recommendation H. 264*, 2003.

[102] Theis and etc. A note on the evaluation of generative models. *arXiv preprint arXiv:1511.01844*, 2015.

[103] Theis and etc. Lossy image compression with compressive autoencoders. *arXiv preprint arXiv:1703.00395*, 2017.

[104] Toderici and etc. Variable rate image compression with recurrent neural networks. *arXiv preprint arXiv:1511.06085*, 2015.

[105] Torfason and etc. Towards image understanding from deep compression without decoding. *arXiv preprint arXiv:1803.06131*, 2018.

[106] Alexander Toshev and Christian Szegedy. Deeppose: Human pose estimation via deep neural networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1653–1660, 2014.

[107] Florian Tramèr, Alexey Kurakin, Nicolas Papernot, Ian Goodfellow, Dan Boneh, and Patrick McDaniel. Ensemble adversarial training: Attacks and defenses. *arXiv preprint arXiv:1705.07204*, 2017.

[108] Jonathan Uesato, Brendan O'Donoghue, Aaron van den Oord, and Pushmeet Kohli. Adversarial risk and the dangers of evaluating against weak attacks. *arXiv preprint arXiv:1802.05666*, 2018.

[109] Gregory K Wallace. The jpeg still picture compression standard. *IEEE transactions on consumer electronics*, 38(1):xviii–xxxiv, 1992.

[110] Jelmer M Wolterink, Tim Leiner, Max A Viergever, and Ivana Išgum. Dilated convolutional neural networks for cardiovascular mr segmentation in congenital heart disease. In *Reconstruction, Segmentation, and Analysis of Medical Images*, pages 95–102. Springer, 2016.

[111] Cihang Xie, Jianyu Wang, Zhishuai Zhang, Zhou Ren, and Alan Yuille. Mitigating adversarial effects through randomization. *arXiv preprint arXiv:1711.01991*, 2017.

[112] Weilin Xu, David Evans, and Yanjun Qi. Feature Squeezing: Detecting Adversarial Examples in Deep Neural Networks. In *Proceedings of the 2018 Network and Distributed Systems Security Symposium (NDSS)*, 2018.

[113] Xiaowei Xu, Yukun Ding, Sharon Xiaobo Hu, Michael Niemier, Jason Cong, Yu Hu, and Yiyu Shi. Scaling for edge inference of deep neural networks. *Nature Electronics*, 1(4):216, 2018.

[114] Xiaowei Xu, Feng Lin, Aosen Wang, Xinwei Yao, Qing Lu, Wenyao Xu, Yiyu Shi, and Yu Hu. Accelerating dynamic time warping with memristor-based customized fabrics. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 37(4):729–741, 2018.

[115] Xiaowei Xu, Feng Lin, Wenyao Xu, Xinwei Yao, Yiyu Shi, Dewen Zeng, and Yu Hu. Mda: A reconfigurable memristor-based distance accelerator for time series mining on data centers. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 2018.

[116] Xiaowei Xu, Qing Lu, Tianchen Wang, Yu Hu, Chen Zhuo, Jinglan Liu, and Yiyu Shi. Efficient hardware implementation of cellular neural networks with incremental quantization and early exit. *ACM Journal on Emerging Technologies in Computing Systems (JETC)*, 14(4):48, 2018.

[117] Xiaowei Xu, Qing Lu, Tianchen Wang, Jinglan Liu, Cheng Zhuo, Xiaobo Sharon Hu, and Yiyu Shi. Edge segmentation: Empowering mobile telemedicine with compressed cellular neural networks. In *Proceedings of the 36th International Conference on Computer-Aided Design*, pages 880–887. IEEE Press, 2017.

[118] Xiaowei Xu, Qing Lu, Lin Yang, Sharon Hu, Danny Chen, Yu Hu, and Yiyu Shi. Quantization of fully convolutional networks for accurate biomedical image segmentation. *Preprint at https://arxiv. org/abs/1803.04907*, 2018.

[119] Xiaowei Xu, Tianchen Wang, Qing Lu, and Yiyu Shi. Resource constrained cellular neural networks for real-time obstacle detection using fpgas. In *2018 19th International Symposium on Quality Electronic Design (ISQED)*, pages 437–440. IEEE, 2018.

[120] Xiaowei Xu, Dewen Zeng, Wenyao Xu, Yiyu Shi, and Yu Hu. An efficient memristor-based distance accelerator for time series data mining on data centers. In *2017 54th ACM/EDAC/IEEE Design Automation Conference (DAC)*, pages 1–6. IEEE, 2017.

[121] Zhongwei Xu, Joan Bartrina-Rapesta, Ian Blanes, Victor Sanchez, Joan Serra-Sagristà, Marcel García-Bach, and Juan Francisco Muñoz. Diagnostically lossless coding of x-ray angiography images based on background suppression. *Computers & Electrical Engineering*, 53:319–332, 2016.

[122] Xue and etc. Segan: Adversarial network with multi-scale l 1 loss for medical image segmentation. *Neuroinformatics*, 16(3-4):383–392, 2018.

[123] Shuiming Ye, Qibin Sun, and Ee-Chien Chang. Detecting digital image forgeries by measuring inconsistencies of blocking artifact. In *Multimedia and Expo, 2007 IEEE International Conference on*, pages 12–15. IEEE, 2007.

[124] Lequan Yu, Jie-Zhi Cheng, Qi Dou, Xin Yang, Hao Chen, Jing Qin, and Pheng-Ann Heng. Automatic 3d cardiovascular mr segmentation with densely-connected volumetric convnets. In *International Conference on Medical Image Computing and Computer-Assisted Intervention*, pages 287–295. Springer, 2017.

[125] Xinfeng Zhang, Shiqi Wang, Ke Gu, Weisi Lin, Siwei Ma, and Wen Gao. Just-noticeable difference-based perceptual optimization for jpeg compression. *IEEE Signal Processing Letters*, 24(1):96–100, 2017.

[126] Tiecheng Zhao, Robert James Taylor, Gang Li, Junnan Wu, and Chunguang Jia. Cloud-based medical image processing system with access control, March 25 2014. US Patent 8,682,049.

VITA

ZIHAO LIU

February 4, 1988	Born, Luoyang, Henan, China
2011-2013	M.Sc., Electrical Engineering Hebei University of Technology Tianjin, China
2013-2015	M.S., Electrical & Computer Engineering Florida International University Miami, Florida
2016-present	Ph.D., Electrical & Computer Engineering Florida International University Miami, Florida

Conference Proceedings

(CVPR-2019) Liu, Zihao, Liu, T., Liu, Q., Xu, N., Lin, X., Wang, Y., Wen, W., (2019). Feature Distillation: DNN-Oriented JPEG Compression Against Adversarial Examples. *In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)(Acceptance Rate 25%).*

(CVPR-2019) Liu, Zihao, Xu, X., Liu, T., Liu, Q., Wang, Y., Shi, Y., et al., (2019). Machine Vision Guided 3D Medical Image Compression for Efficient Transmission and Accurate Segmentation in the Clouds. *In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)(Acceptance Rate 25%).*

(DAC-2018) Liu, Zihao,Liu, T., Wen, W., Jiang, L., Xu, J., Wang, Y., Quan, G., (2018). DeepN-JPEG: a deep neural network favorable JPEG-based image compression framework. *In Proceedings of the 55th Annual Design Automation Conference. Acceptance Rate168/691=24.3%) ACM.*

(DAC-2020) Liu, Q., Liu, T., Liu, Zihao, Wen, W., (2020). Monitoring the Health of Emerging Neural Network Accelerators with Cost-effective Concurrent Test. *In Proceedings of the 57th Annual Design Automation Conference). ACM.*

(DAC-2020) Xu, N., Liu, Q., Liu, T., Liu, Zihao, Wen, W., (2020). Stealing Your Data from Compressed Machine Learning Models. *In Proceedings of the 57th Annual Design Automation Conference). ACM.*

Liu, Zihao, Liu, T., Guo, J., Wu, N., Wen, W., (2018). An ecc-free MLC STT-RAM based approximate memory design for multimedia applications. *In 2018 ieee computer society annual symposium on vlsi (ISVLSI).(Oral Acceptance Rate 29%) IEEE.*

Liu, Q., Liu, T.,Liu, Zihao, Wang, Y., Jin, Y., Wen, W., (2018). Security analysis and enhancement of model compressed deep learning systems under adversarial attacks. *In Proceedings of the 23rd Asia and south pacific design automation conference(Best Paper Award Nomination, 11 out of 271 papers).*

Liu, T.,Liu, Zihao, Liu, Q., Wen, W. (2018). Enhancing the robustness of deep neural-networks from "smart" compression. *In 2018 ieee computer society annual symposium on vlsi (ISVLSI) (pp. 528–532). IEEE.*

Liu, Zihao, Wen, W., Jiang, L., Jin, Y., Quan, G. (2017). A statistical stt-ram retention modelfor fast memory subsystem designs. *In 2017 22nd Asia and south pacific design automation conference(asp-dac)(Acceptance Rate 31%)) (pp. 720–725). IEEE.*

Liu, T., Liu, Zihao, Lin, F., Jin, Y., Quan, G., Wen, W. (2017). Mt-spike: a multi layer time-based spiking neuromorphic architecture with temporal error back propagation. *In Proceedings of the 36th international conference on computer-aided design(Best Paper Nomination from track-Hardware for Embedded Systems, Acceptance rate: 105/399=26%) (pp. 450–457). IEEE.*

(TCAD-2018)Liu, Zihao, Mao, M., Liu, T., Wang, X., Wen, W., Chen, Y., et al. (2018). TriZone: A Design of MLC STT-RAM Cache for Combined Performance, Energy, and Reliability Optimizations. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems (TCAD),37(10), 1985–1998.*

Li, B., Mao, M., Liu, X., Liu, T., Liu, Zihao, Wen, W., Chen, Y. et al. (2019). Thread batching for high-performance energy-efficient gpu memory design. *ACM Journal on Emerging Technologies in Computing (JETC), to appear.*