Learning in Compressed Domains

by

Kai Xu

A Dissertation Presented in Partial Fulfillment
of the Requirements for the Degree
Doctor of Philosophy

Approved February 2021 by the
Graduate Supervisory Committee:

Fengbo Ren, Chair
Baoxin Li
Pavan Turaga
Yezhou Yang

ARIZONA STATE UNIVERSITY

May 2021

ABSTRACT

A massive volume of data is generated at an unprecedented rate in the information age. The growth of data significantly exceeds the computing and storage capacities of the existing digital infrastructure. In the past decade, many methods are invented for data compression, compressive sensing and reconstruction, and compressed learning (learning directly upon compressed data) to overcome the data-explosion challenge. While prior works are predominantly model-based, focus on small models, and not suitable for task-oriented sensing or hardware acceleration, the number of available models for compression-related tasks has escalated by orders of magnitude in the past decade. Motivated by this significant growth and the success of big data, this dissertation proposes to revolutionize both the compressive sensing reconstruction (CSR) and compressed learning (CL) methods from the data-driven perspective.

In this dissertation, a series of topics on data-driven CSR are discussed. Individual data-driven models are proposed for the CSR of bio-signals, images, and videos with improved compression ratio and recovery fidelity trade-off. Specifically, a scalable Laplacian pyramid reconstructive adversarial network (LAPRAN) is proposed for single-image CSR. LAPRAN progressively reconstructs images following the concept of the Laplacian pyramid through the concatenation of multiple reconstructive adversarial networks (RANs). For the CSR of videos, CSVideoNet is proposed to improve the spatial-temporal resolution of reconstructed videos.

Apart from CSR, data-driven CL is discussed in the dissertation. A CL framework is proposed to extract features directly from compressed data for image classification, objection detection, and semantic/instance segmentation. Besides, the spectral bias of neural networks is analyzed from the frequency perspective, leading to a learning-based frequency selection method for identifying the trivial frequency components which can be removed without accuracy loss. Compared with the conventional spatial downsampling approaches, the proposed frequency-domain learning method can achieve higher accuracy with reduced input data size.

The methodologies proposed in this dissertation are not restricted to the above-mentioned applications. The dissertation also discusses other potential applications and directions for future research.

ACKNOWLEDGMENTS

First and foremost, I would like to thank my Ph.D. advisor, Prof. Fengbo Ren, for his sustained guidance and support throughout my Ph.D. life. I am so fortunate to join the PSClab led by Prof. Ren in Summer 2015 as his first Ph.D. student. Prof. Ren is a demonstration of wisdom, reliability, curiosity, and kindness. I really appreciate he gave me the freedom to explore diverse research challenges and forged me as an independent researcher. His critical thoughts and big visions have significantly influenced my view of research, career, and life. I am also thankful for Prof. Baoxin Li, Prof. Pavan Turaga, and Prof. Yezhou Yang for being on my defense committee. I appreciate your insightful suggestions and comments on my dissertation work.

I enjoyed the days working in PSCLab with my previous and current group fellows. I would like to thank Yixing Li, Zhikang Zhang, Michael Riera, Joshua Holloway, Saman Biookaghazadeh, Akshay Dua, Aswin Gunavelu Mohan, Erfan Bank Tavakoli, Zifan Yu for their inspiring discussions and suggestions. I also want to thank my colleague Duo Lv for involving in his marvelous project that inspires me a lot and impacts my career path.

I also want to thank Alibaba Inc., Samsung Research America (SRA), and Kuaishou Inc. for providing me three memorable internship experiences. I would like to thank my amazing co-workers during those three summers: Minghai Qin, Fei Sun, Yuhao Wang, Yen-Kuang Chen, Yongbing Gu, Ao Ren, when I was at Alibaba DAMO Academy, Xiaolong Wang, Dawei Li, Jie Zhang, Boyu Wang, Heming Zhang, Jingwen Zhu, Xianzhi Du, Benwen Zhang, Yang Song when I was SRA and Ning Xu, Yuyan Song when I was at Kuaishou Inc.

I would like to thank the generous donations from NVIDIA and Amazon, as well as the funding support from ASU, Google, and NSF.

Finally, this dissertation is dedicated to my family. I would like to give my sincere thanks to my wife Yuhui Li, my mother Yongfang Jiang, and my father Rongsen Xu for all the years of their love and support.

TABLE OF CONTENTS

APPENDIX

LIST OF TABLES

Table Page

LIST OF FIGURES

Figure Page

Chapter 1

INTRODUCTION

1.1 Motivation

We are living in a world that massive data is generated at an unprecedented rate. According to Gartner's forecast, 14.2 billion devices has been connected to the internet in 2019, and the number will increase to 25 billion by 2021 gar (2018). Meanwhile, five quintillion (ten to the eighteenth) bytes of data are produced every day, and the large-scale data comes from diverse information sources. For example, 1.8 million images are uploaded to social networks every day. Internet-of-Things (IoT) sensors generate and transmit data uninterruptedly and will generate 79.4 zettabytes (ZB) of data in 2025. The growth of data in the present data-explosion era significantly exceeds the capacity of computing and storage devices.

It's a challenging task to solve the data explosion problem, and the challenges lie mainly in twofold factors. First, sensing and acquisition of raw data are difficult. In various emerging applications, it may simply be too costly, or even physically impossible, to build devices capable of acquiring samples at the necessary Nyquist rate. Though compressive sensing (CS) emerges as a promising sensing technique that enables a substantial reduction in the sampling and computation costs, conventional model-based CS reconstruction methods are computationally intensive and not suited for hardware acceleration due to the iterative nature of optimization algorithms. Second, compressed learning is difficult. Many emerging applications with great promise involve the detection of very specific signals and sensing of signal attributes, which requires the system to be fulfilled with the capability of task-oriented sensing. The key to achieve task-oriented sensing is integrating data sensing, compression, and processing into a unified intelligent system. Hence we are motivated to design a system that eliminates the back-and-force of compression-decompression routines and can directly process compressed data in machine learning tasks without data reconstruction.

Reconstruction from Compressed Data. Compressive sensing (CS) is an emerging technique in signal processing as an alternative to traditional Shannon-Nyquist sampling. CS allows a lower sampling rate than the Nyquist sampling rate, and the number of measurements is reduced during acquisition so that additional compression is not required. Although CS enjoys the above advantage, the reconstruction of CS is complicated. Prior model-based methods Becker *et al.* (2011a,b); Dong *et al.* (2014b); Li *et al.* (2009); Metzler *et al.* (2016); Blumensath and Davies (2009); Huggins and Zucker (2007); Tropp and Gilbert (2007) assume the signal is sparse in the time domain, or a transform domain. The above methods suffer from two major drawbacks limiting their practical usage. First, these optimization-based methods are computationally intensive and not suitable for hardware acceleration. Second, the sparsity constraint may not be satisfied for under-researched signals. Even well-studied signals such as natural images do not have an exactly sparse representation on any known basis (DCT, wavelet, or curvelet) Metzler *et al.* (2016). Therefore, the strong dependency on the sparsity constraint usually causes degraded recovery quality in the conventional model-based methods.

The New Trend of Data-Driven Compressive Sensing. While model-based CS methods rely on the sparsity constraint that may not be ubiquitously satisfied, we turn to apply data-driven CS motivated by the recent success of deep neural networks. We propose diverse models for the compressive reconstruction of bio-signals, images, and videos. Compared to the model-based method, the data-driven approach delivers higher reconstruction quality and faster runtime speed.

Learning from Compressed Data. As illustrated above, another prominent challenge for solving the data explosion problem is building a compressed learning system that can directly learn from compressed data. Conventional machine learning systems usually perform a compression, transmission, decompression, and procession routine. In comparison, the compressed learning system eliminates the back-and-force of compression-decompression to save computations. However, prior works suffer from poor performance due to the critical information loss incurred by data compression methods that are not designed for compressed learning. In this work, we propose a frequency-domain compressed learning approach to improve the accuracy in prevailing computer vision tasks such as image classification, object detection, and semantic/instance segmentation. The proposed method delivers a better computation-accuracy trade-off and reduces the communication bandwidth between CPU and GPU, hence suitable for hardware acceleration.

1.2 Dissertation Contributions

As illustrated in Fig. 1.1, We propose models for non-standard compression method, *i.e.*, CS and standard compression such as JPEG. We start by designing a family of deep neural networks for the compressive reconstruction of bio-signals, images, and videos. Then we propose a compressed learning approach that is compatible with the JPEG compression standard. The key contributions of this dissertation can be summarized as follows.

- The first research goal is to enhance the model-based compressive reconstruction methods by introducing the data-driven concept. Specifically, we propose a data-driven CS framework for bio-signals towards improved restricted isometry

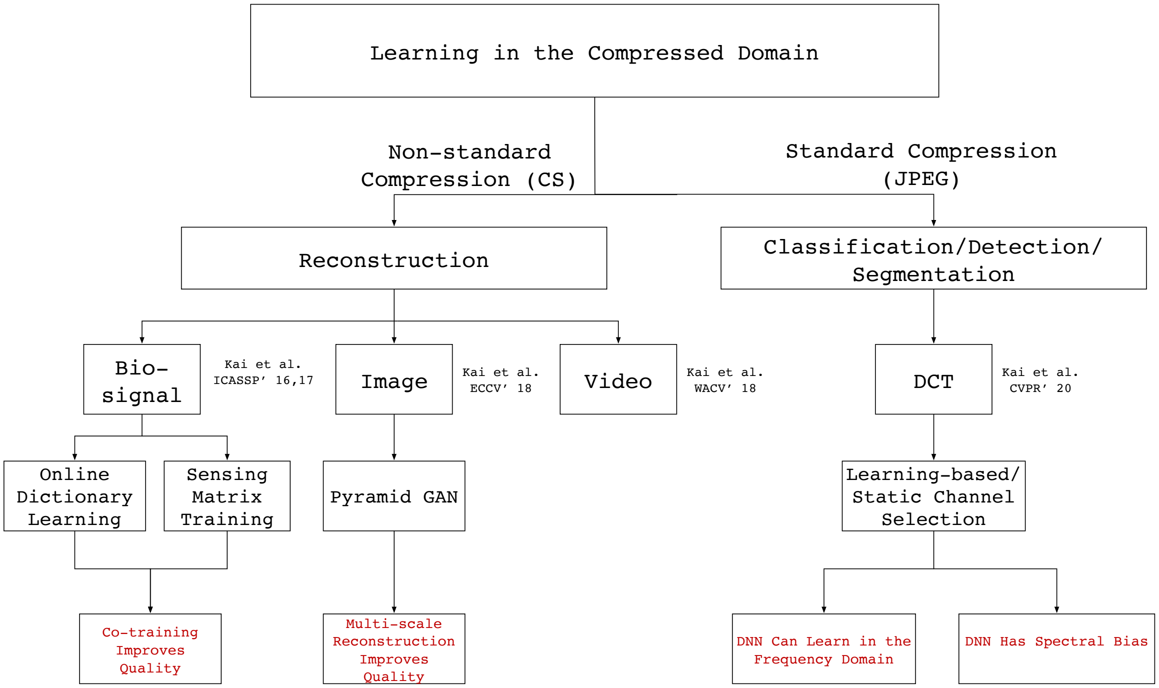

Learning in the Compressed Domain
Non-standard Compression (CS)
Standard Compression (JPEG)
Reconstruction
Classification/Detection/ Segmentation
Bio-signal
Kai et al. ICASSP' 16,17
Image
Kai et al. ECCV' 18
Video
Kai et al. WACV' 18
DCT
Kai et al. CVPR' 20
Online Dictionary Learning
Sensing Matrix Training
Pyramid GAN
Learning-based/ Static Channel Selection
Co-training Improves Quality
Multi-scale Reconstruction Improves Quality
DNN Can Learn in the Frequency Domain
DNN Has Spectral Bias

Figure 1.1: Dissertation overview: proposed architectures for compressive reconstruction and compressed learning.

property (RIP) and signal sparsity, which co-optimizes the sensing matrix and the dictionary by exploiting the intrinsic data structure of bio-signals. The proposed method significantly enhances the reconstruction quality and compression ratio trade-off for the CS of bio-signals.

- The second research mission is to develop an end-to-end data-driven method for the compressive reconstruction of images and videos by leveraging the enormous modeling capacity of deep neural networks. Instead of specifying a strong sparsity assumption on the data used in the model-based methods, the proposed method seeks to avoid any domain-specific presumptions and allows end-to-end compressive reconstruction. Necessary features required for reconstruction are automatically learned by the network itself without any manual interference. Hence, the learned model has better generalization capability and broader application scenarios.

- The third research goal is to develop a compressed learning method from the frequency perspective for task-oriented computer vision tasks. We analyze the spectral bias of neural networks from the frequency perspective and propose a learning-based frequency selection method to identify the trivial frequency components which can be removed without accuracy loss. The proposed method enables efficient learning in the compressed domain and leverages identical structures of the well-known neural networks, such as ResNet-50, MobileNetV2, and Mask R-CNN. Hence, the proposed method can be used as a universal replacement for existing RGB-based computer vision systems.

1.3 Roadmap

In **chapter 2**, we proposed a general framework that utilizes compressive sensing and online dictionary learning simultaneously. The introduction of the online dictionary learning technique produces a dictionary that carries individual characteristics of the original signal. The produced signal has an even sparser representation compared to pre-determined dictionaries. We also demonstrate the data dimension are effectively reduced because of the learned dictionary. The content of this chapter is based primarily on Xu *et al.* (2016).

Chapter 3 extends chapter 2 by introducing a data-driven CS framework that can learn signal characteristics and personalized features from any individual recording of physiologic signals to enhance CS performance with a minimized number of measurements. Such improvements are accomplished by a co-training approach that optimizes the sensing matrix and the dictionary towards improved restricted isometry property and signal sparsity, respectively. The content of this chapter is based primarily on Xu *et al.* (2017).

Chapter 4 addresses the single-image compressive sensing (CS) and reconstruction problem. We propose a scalable Laplacian pyramid reconstructive adversarial network (LAPRAN) that enables high-fidelity, flexible, and fast CS images reconstruction. LAPRAN progressively reconstructs an image following the Laplacian pyramid concept through multiple stages of reconstructive adversarial networks (RANs). At each pyramid level, CS measurements are fused with a contextual latent vector to generate a high-frequency image residual. We demonstrate that LAPRAN can produce hierarchies of reconstructed images and each with an incremental resolution and improved quality. The scalable pyramid structure of LAPRAN enables high-fidelity CS reconstruction with a flexible resolution that is adaptive to a wide range of compression ratios (CRs), which is infeasible with existing methods. The content of this chapter is based primarily on XU *et al.* (2018a).

Chapter 5 addresses the real-time encoding-decoding problem for high-frame-rate video compressive sensing (CS). Unlike prior works that perform reconstruction using iterative optimization-based approaches, we propose a non-iterative model, named "CSVideoNet", which directly learns the inverse mapping of CS and reconstructs the original input in a single forward propagation. To overcome the limitations of existing CS cameras, we propose a multi-rate CNN and a synthesizing RNN to improve the trade-off between compression ratio (CR) and spatial-temporal resolution of the reconstructed videos. We also show that due to the feedforward and high-data-concurrency natures of CSVideoNet, it can take advantage of GPU acceleration to achieve three orders of magnitude speed-up over conventional iterative-based approaches. The content of this chapter is based primarily on Xu and Ren (2018).

In **chapter 6**, we analyze the spectral bias from the frequency perspective and propose a learning-based frequency selection method to identify the trivial frequency components which can be removed without accuracy loss. The proposed learning method leverages identical structures of the well-known neural networks, such as ResNet-50, MobileNetV2, and Mask R-CNN, while accepting the frequency-domain information as the input. We illustrate that compared to the conventional downsampling operation in dealing with high-resolution images, learning in the frequency domain with static channel selection can achieve higher accuracy than the conventional spatial downsampling approach and further reduce the input data size. The content of this chapter is based primarily on Xu *et al.* (2020).

In **Chapter 7**, we summarize the dissertation and provide insights for future research and discuss a broad range of the proposed method of learning in the compressed domain.

Chapter 2

DATA-DRIVEN ONLINE DICTIONARY LEARNING FOR COMPRESSIVE SENSING

2.1 Introduction

The existing heathcare model of the medical system is based on episodic examination or short-term monitoring for disease diagnosis and treatment. The major issues in such a system are the overlook of individual variability and the lack of personal baseline data, due to limited frequency of measurements. Continuous or non-intermittent monitoring is the key to create big data of individual health record for studying the variability and obtaining the personal baseline. Recent advancements in wireless body area networks (WBAN) and bio-sensing techniques has enabled the emergence of miniaturized, non-invasive, cost-effective wireless sensor nodes (WSNs) that can be deployed on the human body for personal health and clinical monitoring Mamaghanian *et al.* (2011). Through WBAN, the monitored data can be transmitted to a near-field mobile aggregator for on-site processing. Through Internet infrastructures, the data can be uploaded to remote servers for storage and data analysis. These technology advancements will eventually transform the existing model of health related services to continuous monitoring for disease prediction and prevention Varshney (2007). Such a wireless health revolution will make healthcare systems more effective and economic, benefiting billions of individuals and the society they live in.

One of the key challenges faced by the long-term wireless health monitoring is the energy efficiency of sensing and information transfer. Due to the limited battery capacity of WSNs, continuous sensing inevitably increases the frequency of battery recharging or replacement, making it less convenient for practical usage. In the WSNs for bio-sensing applications, the energy cost of wireless transmission is about 2 orders of magnitude greater than other operations (e.g., analog-to-digital conversion (ADC)). State-of-the-art radio transmitters exhibit energy efficiency in the nJ/bit range while every other component consumes at most tens of pJ/bit Chen *et al.* (2012). Therefore, reducing the data size for information transfer is the key to improve energy efficiency.

The CS framework Mamaghanian *et al.* (2011); Wang *et al.* (2015) offers a universal and simple data encoding scheme that can compress a variety of physiological signals, providing a viable solution to realizing energy-efficient WSNs for long-term wireless health monitoring. However, the compression ratio (CR) demonstrated by existing frameworks is limited given a signal recovery quality required for diagonosis purposes. In Ansari-Ram and Hosseini-Khayat (2012); Chae *et al.* (2013), percent root-mean-square difference (PRD) of 8.5% and 9% is reported at a CR of 5x and 2.5x for ECG signals, respectively. These frameworks all deal with the sparsity of physiological signals on pre-determined bases and fail to take into account the individual variability in signals that is critical to exact signal recovery.

In this work, we propose an energy-efficient data acquisition framework, customized for the long-term electrocardiogram (ECG) monitoring, which exploits online dictionary learning (ODL) on server nodes to train personalized bases that capture the individual variability for further improving the sparsity of ECG signals. By incorporating such prior knowledge into signal recovery, the CS performance in terms of accuracy-CR trade-off is significantly enhanced, leading to further data size reduction and energy saving on sensor nodes. Additionally, the proposed framework does not require any pre-processing stages on sensor nodes. Alternatively, high reconstruction quality is enforced by pre-processing training data prior to the dictionary learning stage, to eliminate the impact of noise and interference on trained bases, enabling simpler and more cost-effective sensor structures. Experimental results based on MIT-BIH database show that our framework is able to achieve an average PRD of 9% at a CR of 10x. This indicates that our framework can achieve 2-4x additional energy saving on sensor nodes (for the same reconstruction quality) compared to the reference designs Mamaghanian *et al.* (2011); Ansari-Ram and Hosseini-Khayat (2012); Chae *et al.* (2013); Casson and Rodriguez-Villegas (2012). Due to the training and personalization of the dictionary, the proposed framework has the potential to be generally applied to a wide range of physiological signals.

2.2 Preliminaries

2.2.1 Compressive Sensing Review

Assuming a signal $\mathbf{f} \in \mathbb{R}^n$ can be well represented by a sparse vector $\mathbf{x} \in \mathbb{R}^k$ on a certain basis $\mathbf{\Psi} \in \mathbb{R}^{n\times k}$ as $\mathbf{f} = \mathbf{\Psi x}$, then the signal information can be well preserved by projecting $\mathbf{f}$ onto a random domain through a sensing matrix $\mathbf{\Phi} \in \mathbb{R}^{m\times n}$ (m<n) Candès and Wakin (2008), given as

$$\mathbf{y} = \mathbf{\Phi f} = \mathbf{\Phi\Psi x}. \tag{2.1}$$

Candes and et al. Candès *et al.* (2006a) has proven that one has a high probability to recover the sparse coefficient $\mathbf{x}$ by solving the basis pursuit (BP) problem defined as

$$\min_{\mathbf{x}\in\mathbb{R}^k} \|\mathbf{x}\|_1 \quad s.t. \quad \|\mathbf{y} - \mathbf{\Phi\Psi x}\|_2 \leq \varepsilon, \tag{2.2}$$

where ε is an error tolerance term for enhancing the accuracy of the solution considering noise.

2.2.2 Dictionary Learning

Learning dictionaries from data instead of using off-the-shelf bases has been proved effective in improving signal reconstruction performance for images Elad and Aharon (2006). The most recent dictionary learning algorithms Aharon *et al.* (2006); Bruno A. Olshausen (1997); Lee *et al.* (2007) are second-order iterative batch procedures that access the whole training set at each iteration in order to minimize a cost function under certain constraints. Although these algorithms have been shown experimentally faster than first-order gradient descent methods, they cannot effectively handle very large training sets Bottou and Bousquet (2008), because of the involved matrix factorization upon the entire training data. To be able to deal with large data sets for long-term monitoring, the ODL algorithm is adopted in our framework. Compared to the methods mentioned above, ODL has a higher training speed and requires less storage space Mairal *et al.* (2009) because of its elimination of large matrix factorizations. With ODL, it is possible to add new features into the dictionary without stalling the reconstruction, which offers a mechanic of amelioration when a distinctive input is received.

2.2.3 Online Dictionary Learning (ODL)

Assuming the training set is composed of i.i.d. samples following a distribution $p(x)$, ODL draws one sample x_t at a time and alternates between the sparse coding stage and dictionary update stage.

Sparse Coding

The sparse coding problem is a l_1-regularized least-squares problem defined as

$$\alpha_t = \arg\min_{\alpha \in \mathbb{R}^n} \frac{1}{2}\|\mathbf{x}_t - \mathbf{D}_{t-1}\alpha\|_2^2 + \lambda\|\alpha\|_1. \tag{2.3}$$

Due to the high correlations between columns of the dictionary, a Cholesky-based implementation of the LARS-Lasso algorithm, which provides the whole regularization path, is chosen here to solve the sparse coding problem Mairal *et al.* (2010).

Dictionary Updating

At this stage, the objective is to find a dictionary $\mathbf{D}$ that satisfies:

$$\mathbf{D}_t = \arg\min_{\mathbf{D}} \frac{1}{t}\sum_{i=1}^{t} \frac{1}{2}\|\mathbf{x}_i - \mathbf{D}\alpha_i\|_2^2 + \lambda\|\alpha_i\|_1. \tag{2.4}$$

The problem in (4) can be solved by the block coordinate descent algorithm Mairal *et al.* (2009). Overall, the detailed procedure for ODL algorithm is summarized in Algorithm 1.

Algorithm 1 Pseudocode for ODL

```
Input: Input data $x \in \mathbb{R}^n \sim p(x)$, initial dictionary $\mathbf{D_0} \in \mathbb{R}^{n\times k}$, number of iterations t.
Output: Learned dictionary $\mathbf{D}_t$.
Steps:
    Set $\mathbf{A}_0 \leftarrow 0, \mathbf{B}_0 \leftarrow 0$.
    For t=1:T
        Draw a new sample $\mathbf{x}_t$ from $p(x)$.
        Sparse coding: find a sparse coefficient of $\mathbf{x}_t$ under current dictionary $\mathbf{D}_{t-1}$.
        $\mathbf{A}_t \leftarrow \mathbf{A}_{t-1} + \alpha_t\alpha_t^T$.
        $\mathbf{B}_t \leftarrow \mathbf{B}_{t-1} + \mathbf{x}_t\alpha_t^T$.
        Dictionary update: update dictionary $\mathbf{D}_{t-1}$ column by column, the j-th column is given by
        For j=1:k
            $\mathbf{d}_j \leftarrow \frac{1}{(A_t)_{jj}}(\mathbf{B}_t(:,j) - \mathbf{D}\mathbf{A}_t(:,j)) + \mathbf{d}_j$.
            if $\|\mathbf{d}_j\|_2 > 1$, then normalize it to unit form.
        end for
    end for
    Return $\mathbf{D}_t$.
```

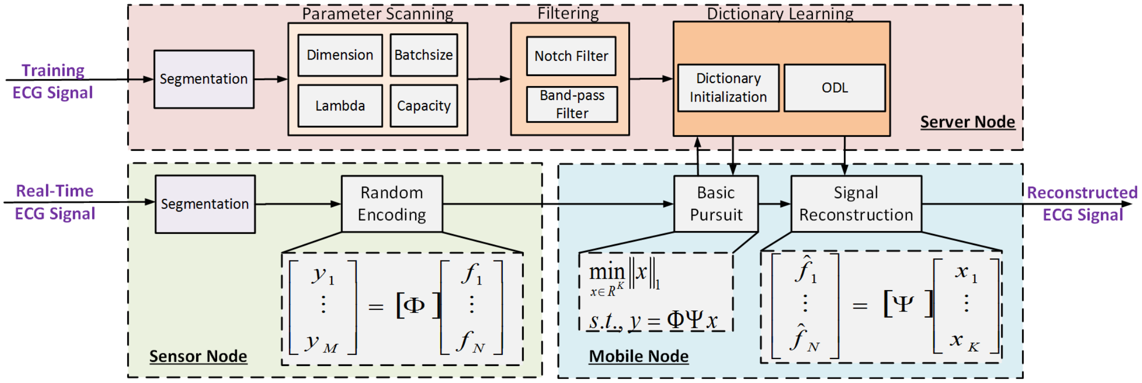


Figure 2.1: Block diagram of the proposed framework. The parameter sweeping and dictionary training procedure are executed on servers. The reconstruction process is performed on mobile platform for providing timely feedback. The random encoding process using random Bernoulli matrix is embedded into the sensor node for effective data compression and energy saving.

2.3 The Proposed Framework

The most recent frameworks on ECG monitoring Lee *et al.* (2014b); Polania *et al.* (2011); Abo-Zahhad *et al.* (2015) adopt a QRS detection process, such as the Pan-Tompkins algorithm, prior to the sensing stage in order to locate the period information of ECG signals. However, integrating the QRS detection process into the sensor nodes not only occupying CPU cycles but also burning excessive power. For wearable applications, an energy-efficient framework must get rid of such pre-processing stages on sensor nodes.

The block diagram of the proposed framework is shown in Fig. 1. It is composed of three functional modules (i.e., dictionary learning, random encoding, and CS signal reconstruction, performed on a server node, a sensor node, and a mobile node, respectively).

The dictionary learning module is used to train personalized bases to capture the individual variability that is critical to exact signal recovery. As dictionary learning directly extracts features from the segmented raw data, the learned dictionary contains critical temporal and spatial information needed for reconstruction. As a result, there is hardly a need for signal alignment. To search for an optimum setup, we first sweep each parameter used in dictionary learning, including signal dimension, batch size for training, regularization coefficient, and dictionary size. The derived parameters are then applied to the dictionary learning module. As the reconstructed signals are the linear composition of atoms in the trained dictionary, a "clean" dictionary thereby have the denoising effect on signal reconstruction. To get a "clean" dictionary, the training data is first filtered by a notch filter to remove power-line inference. Then the signal is passed through a band-pass filter to remove baseline wandering and high-frequency inference. Enabled by the pre-processing in the dictionary learning stage, the proposed framework eliminates the need of employing complicated pre-processing methods prior to random encoding on the sensor node. Instead, a simple segmentation module is sufficient for clean reconstruction.

The initialization in dictionary learning is important. A poorly initialized dictionary may contain bad atoms that are never used Mairal *et al.* (2009). Generally, the dictionary can be initialized by random numbers or input data. For more difficult and regularized problem, it is preferable to start from a less regularized case and gradually increase the regularization coefficients. In our framework, the dictionary is initialized by randomly chosen columns from the input data set for simplicity.

The most notable advantage of ODL over other dictionary learning algorithms, such as K-SVD, is that ODL does not rely on the matrix factorization upon the

Table 2.1: Performance comparison of CS frameworks.

Framework	CR	PRD (%)
Proposed	10	9
Ansari-Ram et al. Ansari-Ram and Hosseini-Khayat (2012)	5	9
Casson el al. Casson and Rodriguez-Villegas (2012)	4	9
Mamaghanian el al. Mamaghanian *et al.* (2011)	3.4	9
Chae et al. Chae *et al.* (2013)	2.5	9

entire training data. As a result, the time cost is much less compared to the non-online versions when handling large training datasets. So a specific input ECG signal that carries new features, such as disease information, can be quickly processed by the dictionary learning module to update the dictionary when necessary. As dictionary update does not depend on the previous samples, the framework also eliminates the demand of large storage space for prior inputs.

BP algorithm, running on the mobile node, is used in our framework to reconstruct high-quality signals. As ODL is compatible with other reconstruction algorithms, more computation efficient algorithms (e.g., fast iterative shrinkage-thresholding algorithm (FISTA) can be implemented to improve accuracy-complexity trade-off).

Experiments are conducted to compare the performance of the proposed framework in terms of recovery quality and CR with the conventional CS frameworks adopting pre-determined basis for the reconstruction of ECG signal. All frameworks employs the same random Bernoulli matrix $\mathbf{\Phi}$ (0/1 only) as the sensing matrix, so the hardware cost of the acquisition module, i.e., the sensor nodes, are the same.

2.3.1 Performance Metrics

The compression ratio (CR) and percent root-mean-square difference (PRD) are used as the performance metrics.

1) Compression Ratio (CR): CR is a measurement of the reduction of the data required to represent the original signal $\mathbf{f}$. If m measurements are required to recover the signal with dimension n, then

$$CR = \frac{n}{m}. \tag{2.5}$$

2) Percent Root-mean-square Difference (PRD): PRD is a measurement of the difference between the original signal $\mathbf{f}$ and the reconstructed signal $\mathbf{f}'$. As arbitrarily low PRD can be achieved by selecting a high DC level in signal $\mathbf{f}$, a more appropriate metric is to remove the DC bias in signal $\mathbf{f}$ as

$$PRD = \frac{\|\mathbf{f} - \mathbf{f}'\|_2}{\|\mathbf{f} - \bar{\mathbf{f}}\|_2} \times 100, \tag{2.6}$$

where $\bar{\mathbf{f}}$ is the mean of signal $\mathbf{f}$.

2.3.2 Experiment Settings and Results

Through parameter sweeping, the dimension of the signal n is set to 256, size of the dictionary k is set to 512. Experiments are carried out based on the MIT-BIH Arrhythmia Database. In the experiments, 649984 samples are divided into 2539 epochs. Each epoch contains 256 samples. Among all the data sets, 512 epochs are randomly chosen to initialize the dictionary, 1621 epochs are used to train the dictionary, and the remaining is used as the testing set. For performance comparison, the pre-determined basis used in the reference framework is a joint basis composed by both discrete cosine transform (DCT) and descrete wavelet transform (DWT) bases Ren and Markovic (2015). This is because the periods components (e.g. QS waves)

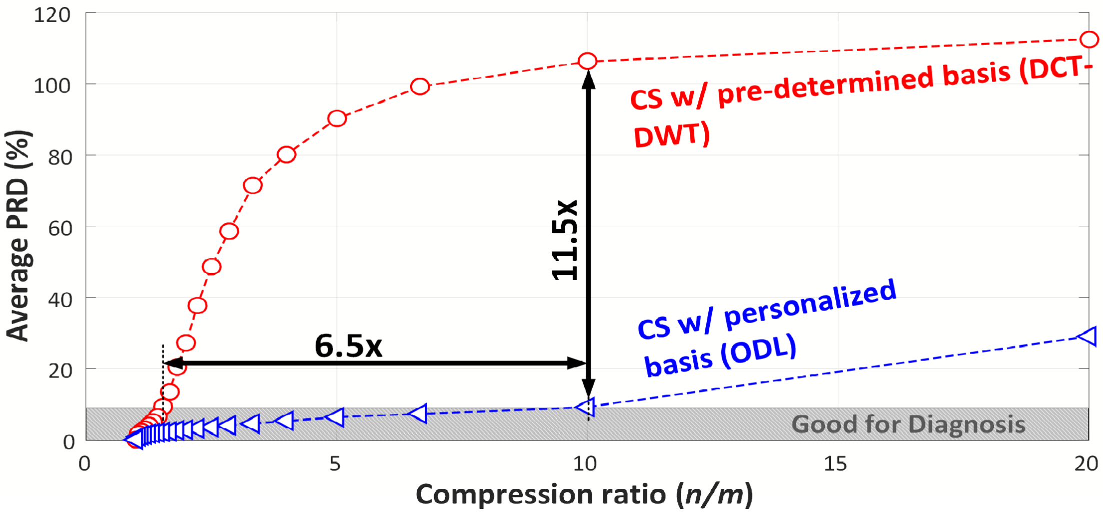


Figure 2.2: Comparison of our proposed framework with conventional CS framework in term of CR.

and the spike components (e.g. R wave) have sparse representations on DCT and DWT basis, respectively.

Figure 2 shows the performance comparison results. Overall, the proposed framework outperform the reference framework significantly due to the use of personlized basis in reconstruction . Specifically, an average PRD of 9%, required for diagnosis purposes Zigel *et al.* (2000), can achieved at a high CR of 10x. This represents a 6.5x more sample size reduction (engergy saving) than the reference framework Ren and Markovic (2015). Table 1 compares the proposed framework with existing CS frameworks Mamaghanian *et al.* (2011); Ansari-Ram and Hosseini-Khayat (2012); Chae *et al.* (2013); Casson and Rodriguez-Villegas (2012) that adopt pre-determined basis in signal recovery. In general, our framework is able to further improve the CR by 2-4x for achieving an average PRD of 9%. Fig.3 demonstrates the high reconstruction quality of the proposed framework in comparison to the reference framework Ren and Markovic (2015) when CR=10.

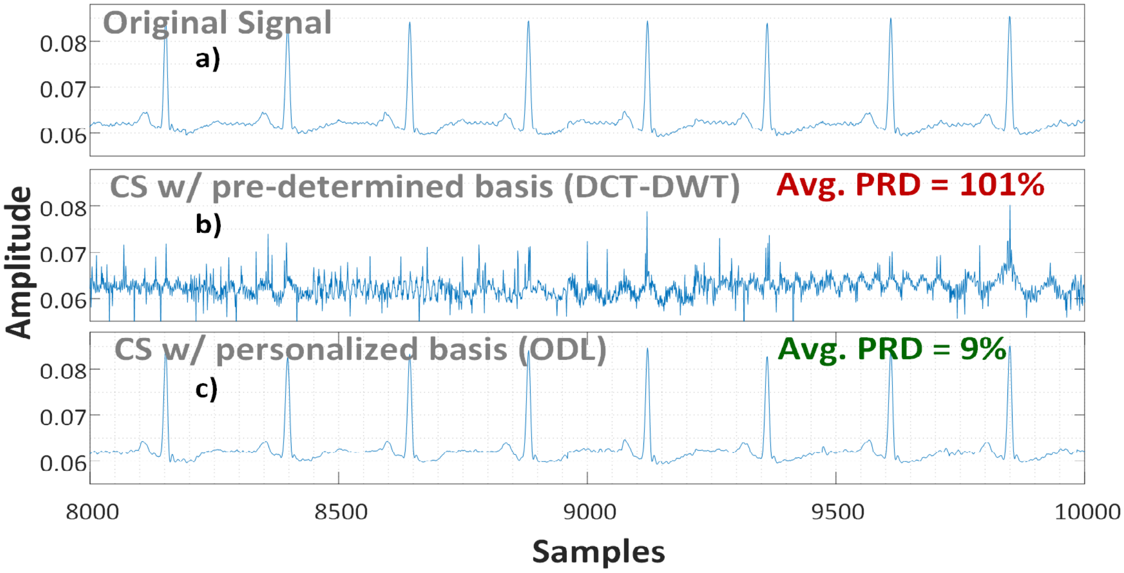


Figure 2.3: Reconstruction result for a segment of ECG signal when CR=10. (a) Original ECG signal; (b) Reconstructed signal using pre-determined DCT-DWT joint basis; (c) Reconstructed signal using online trained dictionary.

2.4 Conclusions

In this chapter, we propose an energy-efficient data acquisition framework combining the notion of CS and ODL for long-term ECG monitoring. The framework significantly enhances CS performance by learning personalized basis to inform signal recovery. Experiment results show that by moving pre-processing to the dictionary learning stage, a simple segmentation process in the sensor nodes is sufficient to recover high-quality signals. In the future work, we will add sub-basis onto which the abnormal ECG signal is projected, when the "healthy" sub-basis is unable to model the original signal accurately.

Chapter 3

DATA-DRIVEN CO-OPTIMIZATION OF COMPRESSIVE SENSING MATRIX AND DICTIONARY

3.1 Introduction

As illustrated in Chapter 2, compressive sensing (CS) offers a universal and straightforward data encoding scheme that can compress a variety of physiological signals, providing a promising solution to the problem. However, most existing CS frameworks are model-driven and suffer from very limited performance when dealing with physiological signals Polania *et al.* (2011); Abo-Zahhad *et al.* (2015); Ren and Markovic (2015). The reasons are two-fold. First, conventional CS frameworks employ random Gaussian or Bernoulli sensing matrices that are generated independently from any data, thereby they fail to leverage any particular geometric structure embedded in the signals of interest. This limits the rank of the sensing matrix required for preserving the Restricted Isometry Property (RIP), leading to limited compression ratio (CR). On the other hand, conventional CS frameworks Polania *et al.* (2011); Lee *et al.* (2014a); Abo-Zahhad *et al.* (2015) that adopt predetermined basis for reconstruction underestimate the intricacy of philological signals and overlook the criticality of individual variability to signal fidelity, which results in very limited reconstruction performance especially at high CR Ren and Markovic (2015). Our previous study Xu *et al.* (2016) has shown that learned dictionaries can better approximate the underlying statistical model of input data. Therefore, they can significantly improve the sparsity of physiological signals as well as reconstruction performance.

3.2 Related Work

There have been some recent work on exploiting data structures for compressive sensing Elad (2007); Duarte-Carvajalino and Sapiro (2009); Hegde *et al.* (2015). In Elad (2007), the authors aim to minimize the averaged mutual coherence between sensing matrix and dictionary. The major limitation of this work is that the mutual coherence is not a direct indicator of RIP, so the optimization result is not suitable for sensor applications. In Duarte-Carvajalino and Sapiro (2009), the authors aim to find a sensing matrix $\mathbf{\Phi}$ and a dictionary $\mathbf{\Psi}$ such that the Gram matrix of the product $\mathbf{\Phi\Psi}$ is as close to the identity matrix as possible. The problem is that the Gram matrix can hardly be the identity matrix in practice as $\mathbf{\Psi}$ is usually over-complete, so the result is sub-optimal. In Hegde *et al.* (2015), the authors aim to preserve the pairwise distance between sample vectors. However, since the NuMax formulation minimizes the transformation distortion against the original signal rather than its sparse coefficient, the trained sensing matrix is not compatible with any over-complete dictionaries. Therefore, these existing approaches are not ideally suitable for the CS of physiological signals in wearable sensing applications.

In this work, we propose a data-driven CS framework that co-optimizes the sensing matrix and the dictionary towards improved restricted isometry property (RIP) and signal sparsity, respectively, by exploiting the intrinsic data structure of physiological signals. Specifically, online dictionary learning (ODL) Mairal *et al.* (2010) is first adopted to train a personalized basis that further improves signal sparsity by capturing the characteristics and individual variability of physiological signals. Based on the learned dictionary, a distortion minimization problem is formulated to construct a near-isometry and low-rank sensing matrix to guarantee a satisfactory recovery performance at improved compression ratios. Overall, the proposed framework keeps the promise to significantly enhance the reconstruction quality and CR trade-off for the CS of physiological signals.

The data-driven nature of the proposed CS framework is very appealing because it fills the gap between the massive medical data and how to utilize them to improve the quality of sensing. The key insight from this study is that the sensor energy efficiency can be enhanced by learning the intrinsic signal structures from big data through cost-effective computation on server systems, rather than doing costly circuit-level development. Moreover, the proposed data-driven framework is equally applicable to a variety of physiological signals and has the potential to be consistently improved as more and more data is collected for training.

3.3 Preliminaries

When fully implemented in the digital domain, CS can be considered as a dimensionality reduction technique for signal compression. Assuming a signal $\mathbf{f}$ can be represented by a sparse vector $\theta \in \mathbb{R}^k$ on a certain basis $\Psi \in \mathbb{R}^{n \times k}$, i.e., $\mathbf{f} = \Psi\theta$, the signal information $\mathbf{x}$ can be well preserved by projecting $\mathbf{f}$ onto a low-dimension space through a sensing matrix $\Phi \in \mathbb{R}^{m \times n}$, ($m \leq n$ and Φ should satisfy 3.2), given as

$$\mathbf{y} = \Phi\mathbf{f} + \mathbf{z} = \Phi\Psi\theta + \mathbf{z}, \tag{3.1}$$

where $\mathbf{z}$ is a noise term.

For robust reconstruction, the matrix Φ should satisfy the RIP Candès *et al.* (2006b) for all k-sparse signal $\mathbf{x}$, defined as

$$(1 - \delta_K \|\mathbf{x}\|_2^2 \leq \|\mathbf{\Phi x}\|_2^2 \leq (1 + \delta_K \|\mathbf{x}\|_2^2)). \tag{3.2}$$

When the RIP holds, Φ approximately preserves the Euclidean norm of all k-sparse signals. Then the sparse coefficient can be solved the following ℓ-1 minimization

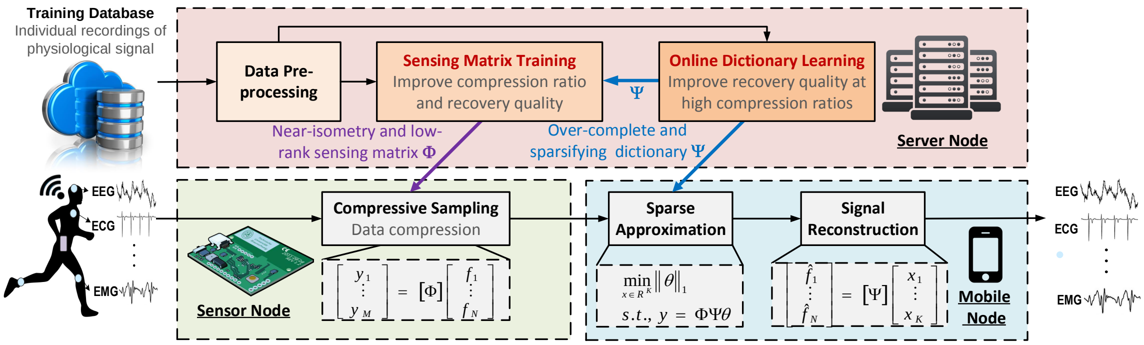

Training Database
Individual recordings of physiological signal
Data Pre-processing
Sensing Matrix Training
Improve compression ratio and recovery quality
Ψ
Online Dictionary Learning
Improve recovery quality at high compression ratios
Server Node
Near-isometry and low-rank sensing matrix Φ
Over-complete and sparsifying dictionary Ψ
EEG
ECG
EMG
Compressive Sampling
Data compression
Sensor Node
$[y_1 \vdots y_M] = [\Phi][f_1 \vdots f_N]$
Sparse Approximation
$\min_{x \in R^K} \|\theta\|_1$ s.t., $y = \Phi\Psi\theta$
Signal Reconstruction
$[\hat{f}_1 \vdots \hat{f}_N] = [\Psi][x_1 \vdots x_K]$
Mobile Node
EEG
ECG
EMG

Figure 3.1: Block diagram of the proposed data-driven compressive sensing framework.

problem with a relaxed constraint,

$$\min_{\theta \in \mathbb{R}^k} \|\theta\|_1 \quad s.t. \quad \|\mathbf{y} - \mathbf{\Phi\Psi}\theta\|_2 \leq \varepsilon. \tag{3.3}$$

If matrix $A \in \mathbb{R}^{m \times n}$ satisfies the RIP of order 2k with $\delta_{2k} < \sqrt{2} - 1$, the solution to 3.3 is equivalent to the original signal with overwhelming probability Candès (2008). In addition, we have

$$\|\mathbf{x}^* - \hat{\mathbf{x}}\|_{\ell 2} \leq C \cdot \frac{\|\mathbf{x} - \mathbf{x_k}\|_{\ell 1}}{\sqrt{k}}, \tag{3.4}$$

where $\mathbf{x} \in \mathbb{R}^n$ is the input signal, $\mathbf{x}_k$ is the k-sparse approximation, and $\hat{\mathbf{x}}$ is the solution to 3.3, and C is a constant which is proportional to the isometry constant δ_{2k}. Eq. 3.4 means a smaller isometry constant guarantees a smaller recovery error, which is suitable for target applications.

3.4 Methodology

3.4.1 Architecture Overview

The architecture of the proposed framework is shown in Fig. 3.1. It is composed of three functional units, including a training unit, a CS sampling unit and signal recovery unit performed on server, sensor and mobile nodes, respectively. Since physiological signals can vary among different patients, a generic basis for all patients usually perform poorly. The dictionary learning module trains personalized basis that captures individual-specific features that are critical to CS recovery, which guarantees a higher sparsity than predetermined basis. Here I employ ODL as the method for dictionary learning. The most notable advantage of ODL is that it does not rely on the matrix factorization upon the entire training data. As a result, the computational complexity is much less compared to the non-online approaches especially for handling large training data. Before ODL is performed, the raw physiological signals must be pre-processed to remove baseline wandering and high-frequency interference. This is essential to achieving a high signal reconstruction quality. Once the dictionary is learned, it can be downloaded to the mobile node to perform accurate signal recovery.

In the proposed framework, the sensing matrix training (SMT) generates a data-specific sensing matrix with minimized rank and a small isometry constant. A small rank further reduces the data size for transmission, and a smaller isometry enhances reconstruction quality denoted by 3.4. Once the sensing matrix is trained, it can be downloaded to the sensor node to perform effective compression of physiological signals for energy-efficient sensing and information transfer.

3.4.2 Sensing Matrix Training (SMT)

Candès and Tao prove that if the sensing matrix $\mathbf{\Phi}$ satisfies the RIP, then ℓ-1 minimization algorithms can successfully recover a sparse signal from noisy measurements Candès *et al.* (2006b). Here I formulate an optimization problem that directly optimizes the RIP towards lower isometry constant δ and lower rank of the sensing matrix $\mathbf{\Phi}$ in 3.5.

$$(1-\delta)\|\theta\|_2 \leq \|\mathbf{\Phi\Psi}\boldsymbol{\theta}\|_2 \leq (1+\delta)\|\theta\|_2, \tag{3.5}$$

where θ is the sparse coefficient vector under the dictionary $\boldsymbol{\Psi}$. 3.5 is equivalent to

$$||\|\boldsymbol{\Phi\Psi\theta}\|_2 - \|\theta\|_2| \leq \delta, \tag{3.6}$$

when θ is normalized.

Suppose we have L sparse coefficients, $\theta_i, i = 1, \ldots, L$, the optimization problem is essentially to guarantee each of them will satisfy 3.6, which can be then reformulated as

$$|\theta_i^T(\boldsymbol{\Psi}^T\boldsymbol{\Phi}^T\boldsymbol{\Phi\Psi} - \boldsymbol{I})\theta_i| \leq \delta, \quad i = 1, \ldots, L. \tag{3.7}$$

Assume $\boldsymbol{A} = \boldsymbol{\Phi\Psi}$, $\mathbf{Y} = \mathbf{A}^T\mathbf{A}$, 3.7 can be represented as

$$|\theta_i^T(\mathbf{Y} - \mathbf{I})\theta_i| \leq \delta, \quad i = 1, \ldots, L. \tag{3.8}$$

As the rank of the sensing matrix implies the data size for transmission after compression, I also aim to minimize the rank of the sensing matrix in 3.9. Since the rank minimization problem is not convex, I use the nuclear norm as a proxy to relax the problem to 3.10.

$$\begin{aligned} \min_{\mathbf{Y}} \quad & \{|\theta_i^T(\mathbf{Y} - \mathbf{I})\theta_i|, \text{ rank}(\mathbf{Y})\}, \quad i = 1, \ldots, L \\ \text{s.t.} \quad & \mathbf{Y} \succeq 0, \\ & \text{diag}(\mathbf{Y}) = [1,\ 1\ , \ldots,\ 1]^T. \end{aligned} \tag{3.9}$$

$$\begin{aligned} \min_{\mathbf{Y}} \quad & (\delta + \beta\|\mathbf{Y}\|^*) \\ \text{s.t.} \quad & \mathbf{Y} \succeq 0, \\ & \text{diag}(\mathbf{Y}) = [1,\ 1\ , \ldots,\ 1]^T \\ & |\theta_i^T(\mathbf{Y} - \mathbf{I})\theta_i| \leq \delta, \quad i = 1, \ldots, L. \end{aligned} \tag{3.10}$$

where β is the penalty parameter for the nuclear norm. Then, I perform an Cholesky decomposition to obtain the matrix $\mathbf{A}$, and a singular value decomposition (SVD) to derive the sensing matrix $\boldsymbol{\Phi}$, as defined in 3.11 and 3.12, respectively.

$$\mathbf{Y} = \mathbf{A}^T\mathbf{A} = \mathbf{U}\mathbf{S}\mathbf{U}^T, \mathbf{A} = (\mathbf{U}\text{sqrt}(\mathbf{S}))^T \tag{3.11}$$

$$\boldsymbol{\Psi} = \mathbf{U}\mathbf{S}\mathbf{V}^T, \boldsymbol{\Psi}^\dagger = \mathbf{V}\mathbf{S}^{-1}\mathbf{U}^T, \boldsymbol{\Phi} = \boldsymbol{A}\boldsymbol{\Psi}^\dagger \tag{3.12}$$

3.4.3 Online Dictionary Learning (ODL)

I seek the dictionary that gives the best representation of every item in the training dataset under the sparsity constraint. The advantage of learning dictionaries from individual recordings of physiological signals is that it provides much better sparse representations than model-driven approaches by exploiting the rich information embedded in the training data. ODL offers faster training speed and fewer storage requirements because of the online processing nature. It is also possible to add new features to the dictionary without stalling the reconstruction using ODL, which offers a mechanic of melioration when a distinctive input is received. Due to the page limit, I would like to refer the readers to Mairal *et al.* (2010) for details of ODL.

3.4.4 Co-training of Sensing Matrix and Dictionary (CTSMD)

I aim to jointly improve signal sparsity and isometry constant through a co-training approach. The proposed CTSMD algorithm is described in Algorithm 1. One should note that the proposed CTSMD algorithm is a non-iterative process. Empirical results show that one round of CTSMD is sufficient to obtain a well-defined results.

Algorithm 1 Pseudocode for CTSMD

```
Input: $\mathbf{x} \in \mathbb{R}^n$, $\boldsymbol{\Psi}_\mathbf{0} \in \mathbb{R}^{n \times k}, \lambda, \beta,$
Output: $\boldsymbol{\Phi}, \boldsymbol{\Psi},$
Online dictionary learning:
```

1) $\theta_t = \arg\min_{\theta \in \mathbb{R}^n} \frac{1}{2}\|\mathbf{x}_t - \boldsymbol{\Psi}_{t-1}\theta\|_2^2 + \lambda\|\theta\|_1,$

2) $\boldsymbol{\Psi}_t = \arg\min_{\boldsymbol{\Psi}} \frac{1}{t}\sum_{i=1}^{t} \frac{1}{2}\|\mathbf{x}_i - \boldsymbol{\Psi}\theta_i\|_2^2 + \lambda\|\theta_i\|_1,$

Sensing matrix training:

3) $\min_{\mathbf{Y}}(\delta + \beta\|\mathbf{Y}\|^*)$

$s.t.\quad \mathbf{Y} \succeq 0,$

$\mathrm{diag}(\mathbf{Y}) = [1\ 1 \ldots\ 1],$

$|\theta_i^T(\mathbf{Y} - \mathbf{I})\theta_i| \leq \delta, \quad i = 1, \ldots, L,$

4)$\mathbf{Y} = \mathbf{A}^T\mathbf{A} = \mathbf{USU}^T, \mathbf{A} = (\mathbf{U}\mathrm{sqrt}(\mathbf{S}))^T,$

5)$\boldsymbol{\Psi} = \mathbf{USV}^T, \boldsymbol{\Psi}^\dagger = \mathbf{VS}^{-1}\mathbf{U}^T, \boldsymbol{\Phi} = \boldsymbol{A}\boldsymbol{\Psi}^\dagger.$

3.5 Experiments

3.5.1 Experiment Setup

Real electrocardiogram (ECG) data from the MIT-BIH arrhythmia ECG database Goldberger *et al.* (2000) is used to benchmark the proposed framework. The customized solver is used for ODL problem and CVX solver Grant and Boyd (2008) is used to solve the SMT problem. Due to the large memory requirement of CVX, our experiments are subjected to limited problem size, which has cost a certain performance degradation across our algorithm. Here we extract 3600 samples, and each sample has a dimension of 128. 3000 and 600 samples are used for training and testing, respectively. The training data is first used with the CTSMD algorithm to construct the sensing matrix and the reconstruction dictionary, which are then used to perform CS measurement and signal reconstruction on the testing data. Three reference approaches are compared in our experiments, i.e. random Gaussian sensing matrix with trained dictionary by ODL, trained sensing matrix by SMT with prede-

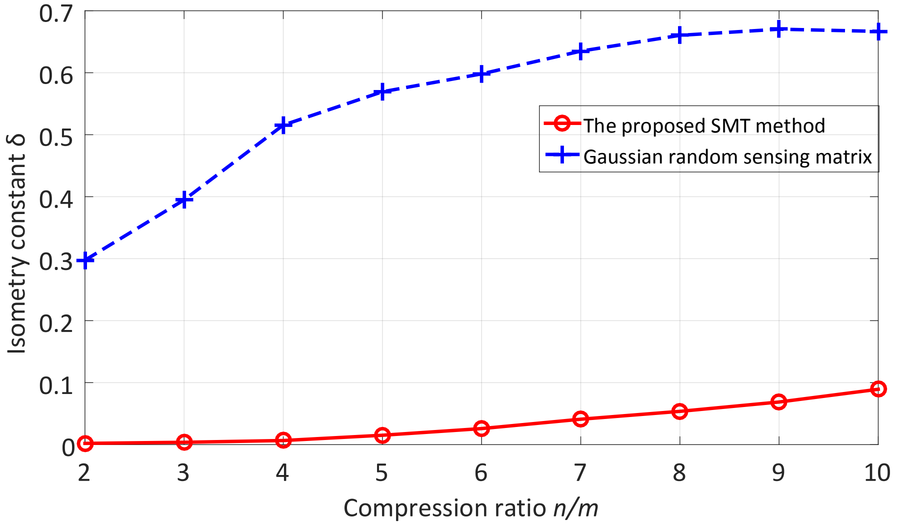


Figure 3.2: Isometry constant under different compression ratios.

termined discrete cosine and wavelet transform (DCT-DWT) dictionary, and random Gaussian sensing matrix with a predetermined DCT-DWT dictionary.

$CR = n/m$ and reconstructed signal-noise ratio (RSNR) $= \|\mathbf{x}\|_2/\|\mathbf{x} - \mathbf{x}'\|_2$ are used as the performance metrics, where n is the dimension of original signal $\mathbf{x}$, m is the number of measurements, and $\mathbf{x}'$ is the reconstructed signal.

3.5.2 Experiment Results

The isometry constant of the trained sensing matrix with respect to CR is shown in Fig. 3.2. Note that the sensing matrices produced by the proposed framework have reduced the isometry constant by over 80% over the Gaussian random matrices across all the CRs. The reduced isometry constant implies better preservation of the signal's geometry structure in the compressed domain. According to 3.4, such improvement will lead to a higher reconstruction accuracy.

The RSNR results at different CR are shown in Fig. 3.3. By using SMT and ODL, RSNR is increased about 5dB and 10dB, respectively. Overall, the proposed

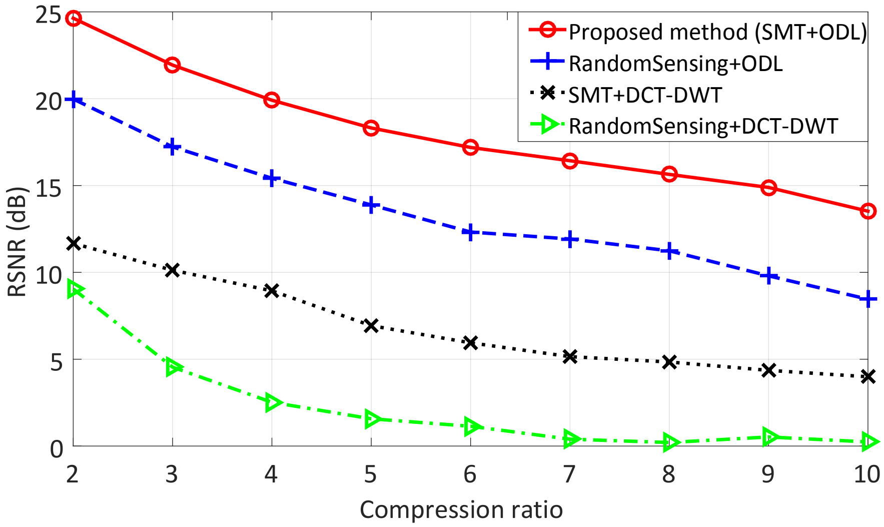


Figure 3.3: RSNR under different compression ratios.

data-driven method achieves a 15dB improvement of RSNR over the model-based approach across all different CRs.

3.6 Conclusion

In this chapter, we propose a data-driven CS framework tailored for the energy-efficient wearable sensing of physiological signals. Exploiting the structure of data is the key to enhancing CS performance. Specifically, the SMT reduces the isometry constant in RIP, and the ODL improves signal sparsity, which are both critical to providing a better recovery performance under improved compression ratios. In future works, we plan to develop customized solver for the SMT problem to handle large dataset. We also need to add binary constraint to SMT for efficient sensor hardware implementations. This will benefit the hardware and energy cost of mobile sensors, which enables the data-driven technique to be used in practical IoTs applications.

Chapter 4

DEEP LEARNING FOR SINGLE IMAGE COMPRESSIVE SENSING RECONSTRUCTION

4.1 Introduction

Compressive sensing (CS) is a transformative sampling technique that is more efficient than Nyquist Sampling. Rather than sampling at the Nyquist rate and then compressing the sampled data, CS aims to directly sense signals in a compressed form while retaining the necessary information for accurate reconstruction. The trade-off for the simplicity of encoding is the intricate reconstruction process. Conventional CS reconstruction algorithms are based on either convex optimization Becker *et al.* (2011a,b); Dong *et al.* (2014b); Li *et al.* (2009); Metzler *et al.* (2016) or greedy/iterative methods Blumensath and Davies (2009); Huggins and Zucker (2007); Tropp and Gilbert (2007). These methods suffer from three major drawbacks limiting their practical usage. First, the iterative nature renders these methods computational intensive and not suitable for hardware acceleration. Second, the widely adopted sparsity constraint assumes the given signal is sparse on a known basis. However, natural images do not have an exactly sparse representation on any known basis (DCT, wavelet, or curvelet) Metzler *et al.* (2016). The strong dependency on the sparsity constraint becomes the performance limiting factor of conventional methods. Constructing over-complete dictionaries with deterministic atoms Xu *et al.* (2016, 2017) can only moderately relax the constraint, as the learned linear sparsity models are often shallow thus have limited impacts. Third, conventional methods have a rigid structure allowing for reconstruction at a fixed resolution only. The recovery quality cannot be guaranteed when the compression ratio (CR) needs to be compromised due to a limited communication bandwidth or storage space. A better solution is to reconstruct at a compromised resolution while keeping a satisfactory reconstruction signal-to-noise ratio (RSNR) rather than dropping the RSNR for a fixed resolution.

Deep neural networks (DNNs) have been explored recently for learning the inverse mapping of CS Dong *et al.* (2016, 2014a); Kim *et al.* (2016); Kulkarni *et al.* (2016). The limitations of existing DNN-based approaches are twofold. First, the reconstruction results tend to be blurry because of the exclusive use of a Euclidean loss. Specifically, the recovery quality of DNN-based methods are usually no better than optimization-based methods when the CR is low, e.g., $CR <= 10$. Second, similar to the optimization-based methods, the existing DNN-based methods all have rigid structures allowing for reconstruction at a fixed and non-adaptive resolution only. The reconstruction will simply fail when the CR is lower than a required threshold.

In this work, I propose a scalable Laplacian pyramid reconstructive adversarial network (LAPRAN) for flexible CS reconstruction that addresses all the problems mentioned above. LAPRAN does not require sparsity as prior knowledge hence can be potentially used in a broader range of applications, especially where the exact signal sparsity model is unknown. When applied to image signals, LAPRAN progressively reconstruct high-fidelity images following the concept of the Laplacian pyramid through multiple stages of specialized reconstructive adversarial networks (RANs). At each pyramid level, CS measurements are fused with a low-dimensional contextual latent vector to generate a reconstructed image with both higher resolution and reconstruction quality. The non-iterative and high-concurrency natures of LAPRAN make it suitable for hardware acceleration. Furthermore, the scalable pyramid structure of LAPRAN enables high-fidelity CS reconstruction with a flexible resolution that can be adaptive to a wide range of CRs. One can dynamically add or remove RAN stages from LAPRAN to reconstruct images at a higher or lower resolution when the CR becomes lower and higher, respectively. Therefore, a consistently superior recovery quality can be guaranteed across a wide range of CRs.

The contributions of this paper are summarized as follows:

- I propose a novel architecture of the neural network model (LAPRAN) that enables high-fidelity, flexible and fast CS reconstruction.

- I propose to fuse CS measurements with contextual latent vectors of low-resolution images at each pyramid level to enhance the CS recovery quality.

- I illustrate that the progressive learning and reconstruction strategy can mitigate the difficulty of the inverse mapping problem in CS. Such a strategy not only accelerates the training by confining the search space but also improves the recovery quality by eliminating the accumulation of errors.

4.2 Related Work

CS reconstruction is inherently an under-determined problem. Prior knowledge, i.e., the structure of signals must be exploited to reduce the information loss after reconstruction. According to the way of applying prior knowledge, CS reconstruction methods can be grouped into three categories: 1) model-based methods, 2) data-driven methods, 3) hybrid methods.

4.2.1 Model-based Reconstruction Methods

Model-based CS reconstruction methods mostly rely on a sparsity prior. For example, basis pursuit (BP), least absolute shrinkage and selection operator (LASSO), and least angle regression (LARS) are all based on ℓ_1 minimization. Other methods exploit other types of prior knowledge to improve the recovery performance. NLR-CS

Dong *et al.* (2014b) proposes a non-local low-rank regularization to exploit the group sparsity of similar patches. TVAL3 Li *et al.* (2009) and EdgeCS Guo and Yin (2010) use a total variation (TV) regularizer to reconstruct sharper images by preserving edges or boundaries more accurately. D-AMP Metzler *et al.* (2016) extends approximate message passing (AMP) to employ denoising algorithms for CS recovery. In general, model-based recovery methods suffer from limited reconstruction quality, especially at high CRs. Because images, though compressible, are not ideally sparse in any commonly used transform domains Metzler *et al.* (2016). Additional knowledge of the image structure is required to further improve the reconstruction quality. Furthermore, when the number of CS measurements available is lower than the theoretical lower bound, the model-based methods will simply fail the reconstruction.

4.2.2 Data-driven Reconstruction Methods

Instead of specifying prior knowledge explicitly, data-driven methods have been explored to learn signal characteristics implicitly. Kuldeep *et al.* and Ali *et al.* propose “ReconNet” Kulkarni *et al.* (2016) and “DeepInverse” Mousavi and Baraniuk (2017), respectively. Both work aims to reconstruct image blocks from CS measurements via convolutional neural networks (CNNs). Experimental results prove that both models are highly robust to noise and able to recover visually better images than the model-based approaches. However, the major drawback of these methods is the exclusive use of the ℓ_2 reconstruction loss for training. As the ℓ_2 loss cannot reliably generate shape images, additional loss metrics must be introduced to further improve the reconstruction quality. In addition, the direct mapping from the low-dimensional measurement domain to the high-dimensional image domain is highly under-determined. This under-determined mapping problem becomes even more notorious as CR increases since the dimension gap between the two domains is enlarged accordingly.

4.2.3 Hybrid Reconstruction Methods

Hybrid methods aim to incorporate the benefits of both model-based and data-driven methods. Such methods first utilize expert knowledge to set up a recovery algorithm and then learn additional knowledge from training data while preserving the model interpretability and performance bounds. Inspired by the denoising-based approximate message passing (D-AMP) algorithm, Chris $etal.$ propose a learned D-AMP (LDAMP) network for CS image reconstruction. The iterative D-AMP algorithm is unrolled and combined with a denoising convolutional neural network (DnCNN) that serves as the denoiser in each iteration. The major drawback of this method is its sophisticated and iterative structure prohibiting parallel training and efficient hardware acceleration.

Inspired by the success of generative adversarial network (GAN) for image generation, Bora *et al.* propose to use a pre-trained DCGAN Radford *et al.* (2015) for CS reconstruction (CSGM) Bora *et al.* (2017). This approach finds a latent vector $\hat{z}$ that minimizes the objective $\|AG(z) - y\|^2$, where G, A and z is the generator, sensing matrix, and CS measurements, respectively. The optimal reconstruction result is represented as $G(\hat{z})$. Differently, the proposed LAPRAN directly synthesize an image from CS measurements, which alleviates the exploration of an additional latent space. Although both approaches are GAN-based, they represent two fundamentally different CS reconstruction schemes. CSGM is a sparse-synthesize model Candès *et al.* (2006b,a) that approximates an unknown signal as $x = G(z)$, where the sparse coefficient (z) is measured concurrently. While LAPRAN is a co-sparse-analysis model Nam *et al.* (2013); Candès *et al.* (2011) that directly synthesize an unknown signal

x from the corresponding CS measurements y according to $x = G(y)$. Hence, I call the building block of the proposed model reconstructive adversarial network (RAN) instead of GAN. RAN elegantly approximates the nature image distribution from CS measurement samples, avoiding the detour in the synthesize model. While multiple network propagations are needed to obtain the optimal $\hat{z}$ in CSGM, LAPRAN finishes reconstruction in a single feedforward propagation. Therefore, LAPRAN has lower computational complexity and a faster reconstruction speed.

4.3 Methodology

The overall structure of the proposed CS system is shown in Figure 4.1. It is composed of two functional units, a multi-rate random encoder for sampling and a LAPRAN for reconstruction. The multi-rate random encoder generates multiple CS measurements with different CRs from a single image. LAPRAN takes the CS measurements as inputs and progressively reconstructs the original image in multiple hierarchies with incremental resolutions and recovery quality. In the first stage, RAN1 reconstructs a low-resolution thumbnail of the original image (8×8). The following RANs at each stage fuses the low-resolution input generated by the previous stage with CS measurements to produce a reconstructed image upsampled by a factor of 2. Therefore, the resolution of the reconstructed image is progressively improved throughout the cascaded RANs. The proposed LAPRAN architecture is highly scalable. One can concatenate more RANs (just like "LEGO" blocks) to gradually increase the resolution of the reconstructed image. Each building block of LAPRAN is detailed below. Further details about the LAPRAN architecture are provided in the supplementary materials.

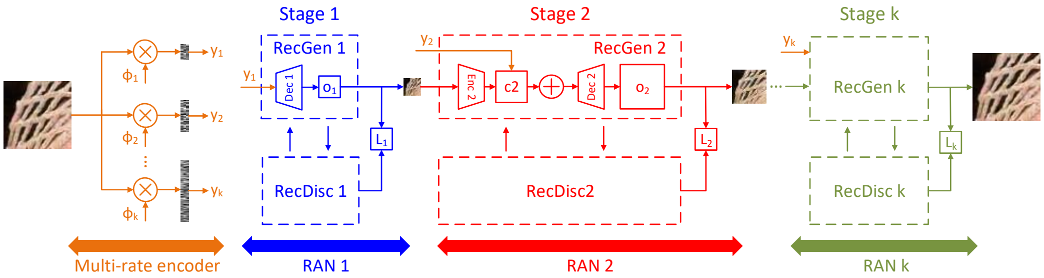


Figure 4.1: Overall structure of the proposed LAPRAN. The CS measurement of a high-dimensional image is performed by a multi-rate random encoder. The LAPRAN takes CS measurements as inputs and progressively reconstructs an original image in multiple hierarchies with incremental resolutions and recovery qualities. At each pyramid level, RAN generates an image residual, which is subsequently combined with an upscaled output from the previous level to form a higher-resolution output of the current level (upsampling and upscaling respectively refers to increasing the image resolution with and without new details added). The detailed structure of RAN is shown in Figure 4.3.

4.3.1 Multi-rate CS Encoder

I propose a multi-rate random encoder for CS sampling. Given an input image, the encoder generates multiple CS measurements $\{\mathbf{y_1}, \cdots, \mathbf{y_t}\}$ simultaneously, each has a different dimension. The generated measurements are fed into each stage of the RANs as input, i.e., $\{\mathbf{y_1}, \cdots, \mathbf{y_k}\}$ is forward to {RAN1, ..., RANk}, respectively. According to the rate-distortion theory Davisson (1972), the minimum bit-rate is positively related to the reconstruction quality, which indicates that the i-th RAN requires more information than all the previous RANs in order to improve the image resolution by adding finer details incrementally. The quantitative analysis of the number of measurements required for each RAN is as follows. Let $\mathbf{A}$ be an $m \times n$ sensing matrix that satisfies the restricted isometry property (RIP) of order $2k$, and the isometry constant is $\delta_{2k} \in (0, \frac{1}{2}]$. According to the CS theory Davenport (2010), the lower bound of the number of CS measurements required for satisfying RIP is defined as: $m \geq Ck\log(\frac{n}{k})$, where $C = \frac{1}{2}\log(\sqrt{24}+1) \approx 0.28$. In the CS image reconstruction problem, let the number of input measurements required by two adjacent RANs for accurately reconstructing a $N \times N$ image and a $2N \times 2N$ image is $m1$ and $m2$, respectively, I define the measurement increment ratio as $\beta = \frac{m2}{m1}$. If I assume the sparsity ratio ($\frac{k}{n}$) of the two images remains constant across the two adjacent RANs, then β can be calculated as:

$$\beta = \frac{4k \times \log[(2N \times 2N)/4k]}{k \times \log[(N \times N)/k]} = 4. \tag{4.1}$$

Equation (4.1) indicates that the number of CS measurements (as well as CR) required for a former RAN should be at least 1/4 of a latter one in order to guarantee a satisfactory reconstruction performance. One should note that $\beta = 4$ is the upper bound, lower β values can be used to offer better reconstruction performance at the cost of collecting more CS measurements in early stages. In this work, I adopt $\beta = 2$ to set a gradually increasing CR at different stages instead of using a unified CR. Since the dimension of a measurement vector equals to the number of rows in a sensing matrix, the k sensing matrices in Figure 4.1 have the following dimensions: $\mathbf{\Phi_1} \in \mathbb{R}^{m \times N}, \mathbf{\Phi_2} \in \mathbb{R}^{\lfloor \beta m \rfloor \times N}, \cdots, \mathbf{\Phi_k} \in \mathbb{R}^{\lfloor \beta^{k-1} m \rfloor \times N}$. An example of the sensing matrix used for the multi-rate encoding of a 4-stage LAPRAN is illustrated in Figure 4.2. The generated measurements $\mathbf{y_1} \in \mathbb{R}^{m}, \mathbf{y_2} \in \mathbb{R}^{2m}, \mathbf{y_3} \in \mathbb{R}^{4m}, \mathbf{y_4} \in \mathbb{R}^{8m}$ is used as the input to RAN1, RAN2, RAN3 and RAN4, respectively. With respect to a k-stage LAPRAN, I only need to generate $\mathbf{y_t}$ for training. Since $\mathbf{y_i}$ is always a subset of $\mathbf{y_{i+1}}$, I can feed the first $\lfloor \beta^{i-1} m \rfloor$ elements of $\mathbf{y_t}$ to the i-th stage in a backward fashion.

The proposed LAPRAN enables CS reconstruction with a flexible resolution, which is not feasible with existing methods. When the number of CS measurements fail to meet the required threshold, the existing methods will fail to reconstruct with no room for maneuver. Alternatively, the proposed method can still reconstruct lower-resolution previews of the image with less detail in the case that the CS measurements are insufficient. The output of each RAN constitutes an image pyramid, providing the

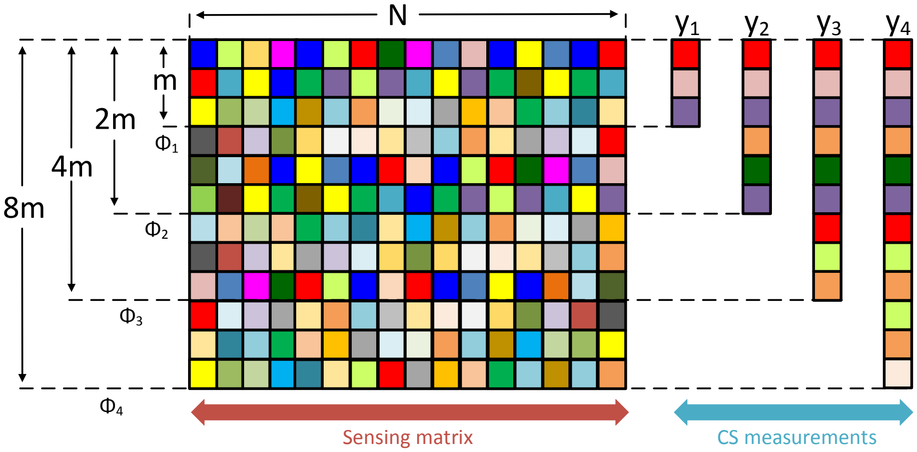


Figure 4.2: Illustration of a sensing matrix for multi-rate CS. The four sensing matrices $\mathbf{\Phi_1} \in \mathbb{R}^{m\times N}$, $\mathbf{\Phi_2} \in \mathbb{R}^{2m\times N}$, $\mathbf{\Phi_3} \in \mathbb{R}^{4m\times N}$, $\mathbf{\Phi_4} \in \mathbb{R}^{8m\times N}$ are used to generate the four CS measurements $\{\mathbf{y_1}, \mathbf{y_2}, \mathbf{y_3}, \mathbf{y_4}\} \in \mathbb{R}^{\{m,2m,4m,8m\}}$. y_1, y_2, y_3, y_4 is fed into RNN1 to RNN4 as the information source, respectively.

user with great flexibility in choosing the desired resolution of reconstructed images.

4.3.2 RAN for CS Image Reconstruction

I propose a RAN at each pyramid level to generate the reconstructed image with a fixed resolution. A RAN is composed of a reconstructive generator denoted as“RecGen”, and a discriminator denoted as “RecDisc.” RecDisc follows the structure of DCGAN Radford *et al.* (2015), and the structure of RecGen is specially customized for reconstruction. Taking RecGen2 in the 2nd RNN stage as an example (see Figure 4.3), $\{\mathbf{i_2}, \mathbf{r_2}, \mathbf{u_2}, \mathbf{o_2}\}$ is the contextual input from the previous stage, image residual, upscaled input, and output image, respectively. $\mathbf{y_2}$ is the input measurements generated by the multi-rate CS encoder. RecGen2 is composed of two branches: 1) the upper branch that generates an upscaled input image $\mathbf{u_2}$ via a deconvolutional neural network (deconv1); and 2) the lower branch that generates an image residual $\mathbf{r_2}$ to compensate for the artifacts introduced by the upper branch. Note that $\mathbf{u_2}$ is upscaled from a lower-resolution image, thus $\mathbf{u_2}$ lacks high-frequency components

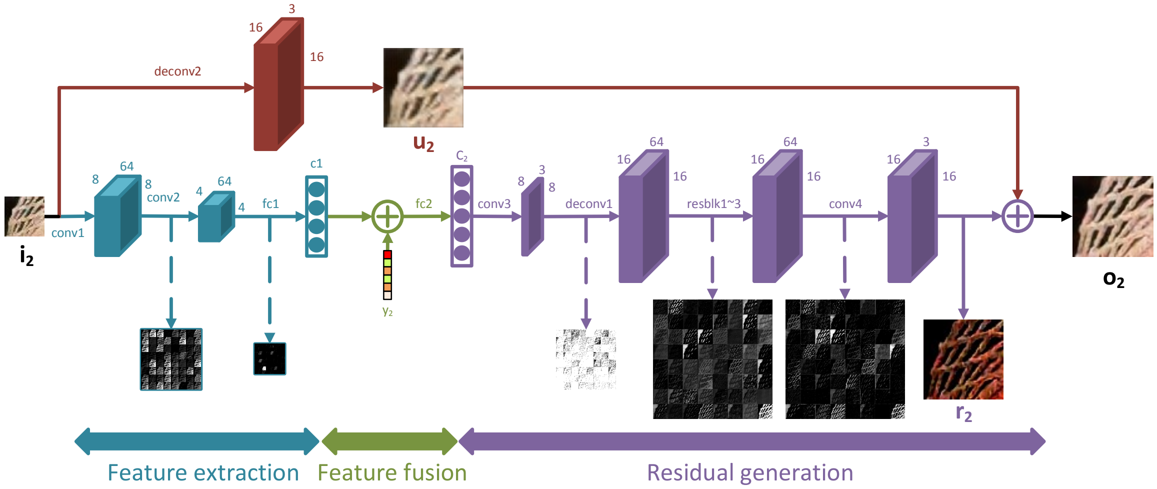


Figure 4.3: The structure of RecGen2. A low-resolution input image $\mathbf{i_2}$ is transformed into a high-frequency image residual $\mathbf{r_2}$ by an encoder-decoder network. A high-resolution output image is generated by adding the image residual to the upscaled input image. The dimension of each feature map is denoted in the figure. An example output of each convolutional layer is also shown.

(see Figure 4.3) and only provides a coarse approximation to the higher-resolution ground-truth image. It is the addition of the high-frequency residual $\mathbf{r_2}$ that recovers the entire frequency range of the image thus substantially improves the reconstruction quality Denton *et al.* (2015).

The input $\mathbf{i_2}$ is treated as a low-resolution context for generating the residual image $\mathbf{r_2}$. I propose to first use an encoder to extract a contextual latent vector $\mathbf{c_1}$ to represent the low-resolution context $\mathbf{i_2}$. The encoder is composed of two convolutional layers and a fully-connected layer. To guarantee an equal contribution to the feature after fusion, the contextual latent vector $\mathbf{c_1}$ has the same dimension as the CS measurement $\mathbf{y_2}$. It should be noted that by increasing the dimension of $\mathbf{c_1}$, one can expect more image patterns coming from the contextual input appear in the final reconstruction, and vice versa. $\mathbf{c_1}$ is fused with the CS measurement $\mathbf{y_2}$ through concatenation (referred to as “early fusion” in Snoek *et al.* (2005)) in a feature space. The fully-connected layer is used to transform the fused vector back to a feature map that has the same dimension as the contextual input $\mathbf{i_2}$. A common practice of upscaling is to use an unpooling layer Zeiler and Fergus (2014) or interpolation layer (bilinear, bicubic, or nearest neighbor). However, these methods are either non-invertible or non-trainable. Instead, I apply a deconvolutional layer deconv1 Zeiler *et al.* (2011) to learn the upsampling of the fused feature map. I set up three residual blocks (resblk1$\sim$3) He *et al.* (2016) to process the upsampled feature map to generate the image residual $\mathbf{r_2}$, which is later combined with $\mathbf{u_2}$ generated by the upper branch (deconv2) to form the final output image.

Learning from context.

Instead of reconstructing the original image from CS measurements directly, I propose to exploit the low-resolution context (i_2 in Figure 4.3) to condition for reconstruction. The proposed conditional reconstruction scheme is fundamentally different from the conventional methods that solely rely on CS measurements. The reason is as follows.

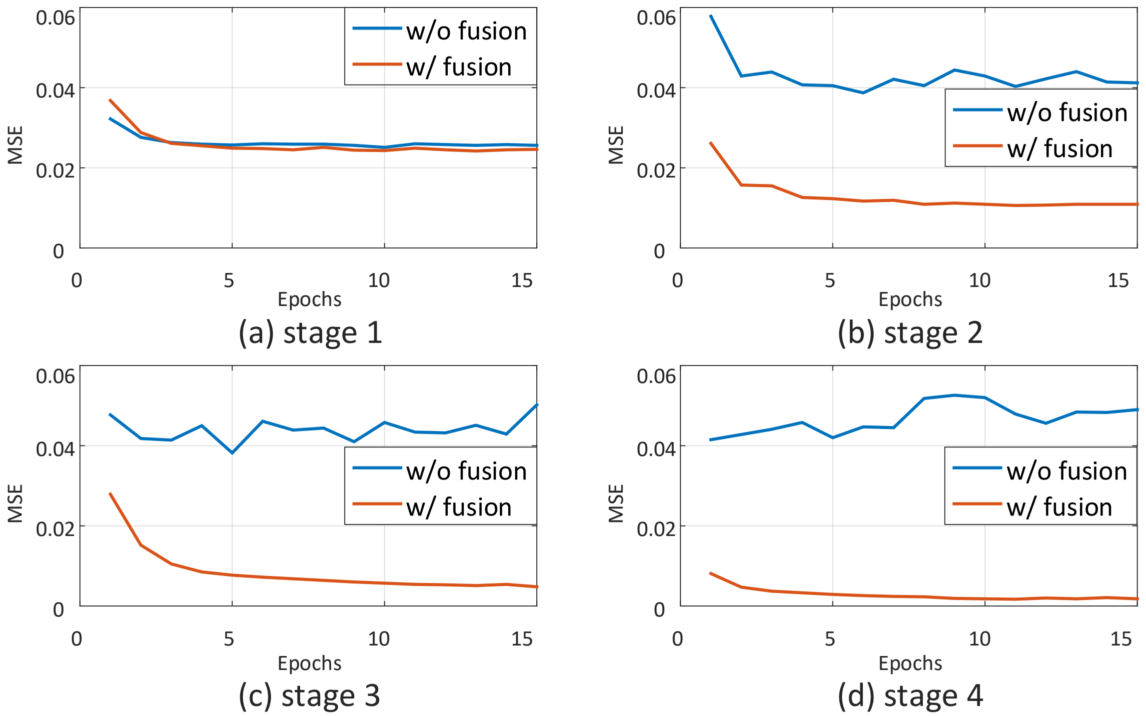
Learning the inverse reconstructive mapping is a highly under-determined problem, hence notoriously tricky to solve. I need to accurately predict each pixel value in such an exceptionally high-dimensional space. All the existing data-driven methods directly search in such a vast space and try to establish a direct mapping from the low-dimensional CS measurements to the high-dimensional ground-truth. The intricacy of the problem and the lack of additional constraints make the search process inefficient and untrustworthy. Differently, I delegate the low-resolution context to confine the sub-search space, i.e., the candidates that are far from the context in the search space will be obviated. Besides, the CS measurements supplement the necessary information needed for recovering the entire frequency spectrum of the image. The fusion of the context and CS measurements hence improve both convergence speed and recovery accuracy.

Residual learning.

In LAPRAN, the RecGen of each RAN is similar to a segment of the ResNet in He *et al.* (2016). All the convolutional layers are followed by a spatial batch normalization (BN) layer Ioffe and Szegedy (2015) and a ReLU except for the output layer. The output layer uses a Tanh activation function to ensure the output image has pixel values in the range of [0, 255]. The use of BN and normalized weight initialization LeCun *et al.* (1998) alleviates the problem of vanishing or exploding gradients hence improve both convergence accuracy and speed.

4.3.3 Cascaded RANs for Flexible CS Reconstruction

The existing DNN-based methods all have rigid structures allowing for reconstruction with a fixed CR and at a non-adaptive resolution only. A new model must be retrained from scratch when a different CR is used in the encoding process. Inspired by the self-similarity based super resolution (SR) method Glasner *et al.* (2009); Cui *et al.* (2014), I propose a flexible CS reconstruction approach realized by dynamically cascading multiple RANs (see Figure 4.1) at runtime. Upon training, each RAN corresponds to a specific resolution of the reconstructed image as well as an upper bound of the CR needed for accurate reconstruction. The thresholds of CR at different stages should be determined from experiments given a target accuracy metric. At runtime, depending on the CR of inputs, only the RANs with a higher CR threshold will be enabled for reconstruction. As a result, the proposed LAPRAN can perform high-fidelity CS reconstruction with a flexible resolution that is adaptive to a wide range of CRs. This merit is particularly significant to the CS application scenarios, where the CR must be adaptive to the dynamic requirements of storage space or communication bandwidth. When the CR is compromised in such an application scenario, all the



(a) stage 1

(b) stage 2

(c) stage 3

(d) stage 4

Figure 4.4: Convergence analysis. I compare the MSE test error using the CIFAR10 dataset at the CR of 10. The results without measurement fusion can be regarded as the performance of an SR approach. The MSE loss of the SR approach cannot be effectively reduced after stage 1 because of the lack of new information.

existing methods will fail the reconstruction, while the proposed LAPRAN can still reconstruct an accurate preview of the image at a reduced resolution.

Another advantage of the proposed LAPRAN is that its hierarchical structure reduces the difficulty of training. CS reconstruction is a highly under-determined problem that has a humongous space for searching. Therefore, it is very challenging for a single network to approximate the inverse mapping accurately. Adopting a divide-and-conquer strategy, I propose to divide a highly under-determined problem into a series of lightly under-determined problems and conquer them in multiple hierarchies. As the dimensionality gap between the input and output in each subproblem is significantly reduced, the difficulty for learning each mapping is much reduced compared to the original problem. Besides, since the hierarchical structure leverage a series of upsampling operations, error accumulation occurs at each stage. To alleviate such a problem, I define a loss function and perform back-propagation per stage independently. The training error is effectively reduced after each stage compared to the case that a single back-propagation is performed at the final output.The injected CS measurements at each pyramid level are the key for CS reconstruction, which distinguishes the proposed approach from image SR methods. The SR models Dong *et al.* (2016, 2014a); Kim *et al.* (2016); Lai *et al.* (2017) are responsible for inferring the high-frequency components non-existed in the input. From the frequency perspective, SR models should be adequately non-linear to compensate for the frequency gap, which inevitably results in complicated structures. Differently, the proposed approach incorporates new information provided by CS measurements into the reconstruction at each stage. The CS measurements supplement necessary information needed for recovering the entire frequency spectrum of an image, which is a powerful information source for learning visual representations. Consequently, both the resolution and the quality of the reconstructed images increase across different stages in the proposed approach. To illustrate this point, I compare LAPRAN with a variant that has no fusion mechanism implemented at each stage (an SR counterpart). The comparison results are shown in Figure 4.4. It is obvious that the reconstruction accuracy of the proposed LAPRAN is consistently improved stage by stage, while the SR counterpart suffers from limited performance improvement.

4.3.4 Reconstruction Loss

I use a pixel-wise ℓ_2 reconstruction loss and an adversarial loss for training. The ℓ_2 loss finds an overall structure of a reconstructed image. The adversarial loss picks up a particular mode from the image distribution and generates a more authentic output Pathak *et al.* (2016). The overall loss function is defined as follows:

$$
\begin{aligned}
\mathbf{z} \sim \mathrm{Enc}(\mathbf{z}|\mathbf{x_l}), \quad \mathbf{x_h} &= G(\mathbf{y}|\mathbf{z}), \\
\mathcal{L}_{adv}(G, D) &= \mathbb{E}_{\mathbf{x_h}}[\log D(\mathbf{x_h}|\mathbf{z})] + \mathbb{E}_{\mathbf{y}}[\log(1 - D(G(\mathbf{y}|\mathbf{z})))], \\
\mathcal{L}_{euc} &= \mathbb{E}_{\mathbf{x_h}}[\|\mathbf{x_h} - \mathbf{x_G}\|_2], \\
\mathcal{L}_{total} &= \lambda_{adv}\mathcal{L}_{adv} + \lambda_{euc}\mathcal{L}_{euc}, \qquad (4.2)
\end{aligned}
$$

where $\mathbf{x_l}$, $\mathbf{x_h}$ and $\mathbf{x_G}, \mathbf{y}$ is the low-resolution input image, the high-resolution output image, the ground-truth image, and the CS measurement, respectively. The encoder function (Enc) maps a low-resolution input $\mathbf{x_l}$ to a distribution over a contextual latent vector $\mathbf{z}$.

4.3.5 Training

The training of each RAN is performed individually and sequentially. I start by training the first stage and the output is used as the input for the second stage. The training of all the subsequent stages is performed in such a sequential fashion. Motivated by the fact that the RANs in different stages share a similar structure but with different output dimensionality, I initialize the training of each stage with the pre-trained weights of the previous stage to take advantage of transfer learning. Such a training scheme is shown in experiments to be more stable and has faster convergence than those with static initialization (such as Gaussian or Xavier). Besides, the weight transfer between adjacent stages helps to tackle the notorious mode collapse problem in GAN since the pre-trained weights already cover the diversity existed in training images. It is recommended to leverage weight transfer to facilitate the training of the remaining RANs.

4.4 Experiments

In this section, we evaluate the performance of the proposed method. We first describe the datasets used for training and testing. Then, the parameters used for training are provided. Finally, we compare our method with state-of-the-art CS reconstruction methods.

4.4.1 Datasets

We train and evaluate the proposed LAPRAN on three widely used benchmark datasets. The first two are MNIST and CIFAR10. The third dataset is made following the rule used in prior SR work Kim *et al.* (2016); Lai *et al.* (2017); Schulter *et al.* (2015), which uses 91 images from Yang et al. Yang *et al.* (2010) and 200 images from the Berkeley Segmentation Dataset (BSD) Arbelaez *et al.* (2011). The 291 images are augmented (rotation and flip) and cut into $228,688$ patches as the training data. Set5 Bevilacqua *et al.* (2012) and Set14 Zeyde *et al.* (2012) are pre-processed using the same method and used for testing.

4.4.2 Training Parameters

We implemented a four-stage LAPRAN for CS image reconstruction. We resize each training image to 64×64 and train the entire LAPRAN with a batch size of 128 for 100 epochs. We use Adam solver with a learning rate of 1×10^{-4}. The training takes roughly two days on a single NVidia Titan X GPU.

4.4.3 Comparisons with State-of-the-art

We compare the proposed LAPRAN with six state-of-the-art CS reconstruction methods: NLR-CS Dong *et al.* (2014b), TVAL3 Li *et al.* (2009), BM3D-AMP (D-

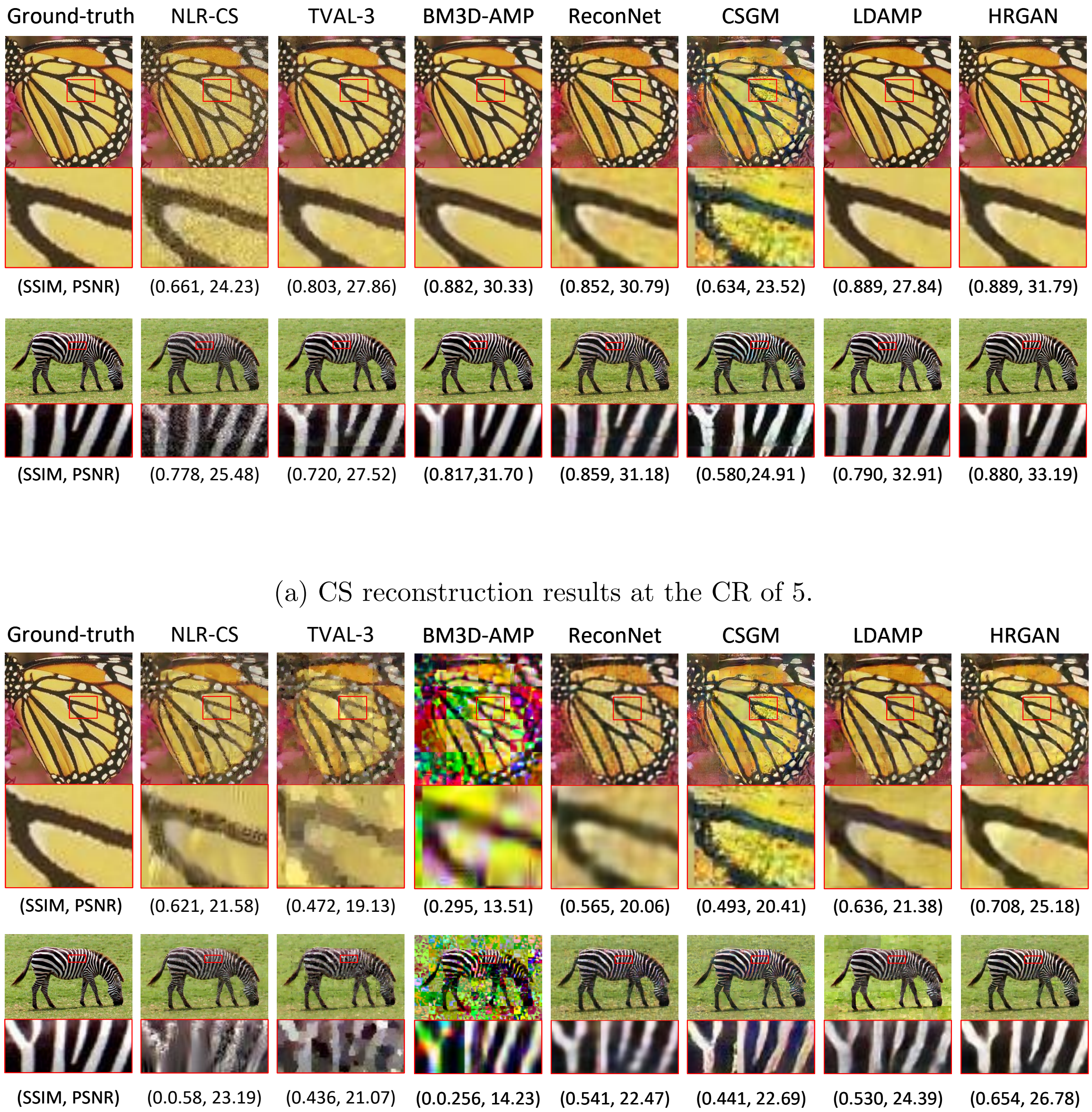
AMP with BM3D denoiser Dabov *et al.* (2007)), ReconNet Kulkarni *et al.* (2016), CS reconstruction using a generative model (CSGM) Bora *et al.* (2017), and learned D-AMPMetzler *et al.* (2017). We summarize the major differences between the proposed and the refernce methods in Table 4.1. Structural similarity (SSIM) and peak signal-to noise ratio (PSNR) are used as the performce metrics in the benchmark.

The quantitative comparison of reconstruction performance is shown in Table 4.2. The proposed LAPRAN achieves the best recovery quality on all the testing datasets and at all CRs. Especially, the performance degradation of the LAPRAN at large CRs ($\geq$20) is well bounded. The main reasons are two-fold. First, our approach adopts a progressive reconstruction strategy that mitigates the difficulty of approximating the inverse mapping of CS. In contrast, CSGM tries to generate high-resolution images in a single step thus has a low reconstruction quality due to the difficulty in learning. Second, our approach utilizes a low-resolution image as input to guide the generation process at each stage, which helps to further reduce the search space of the under-determined problem by eliminating irrelevant candidates. The visual comparison of reconstructed images (at the CRs of 5 and 20) from Set 5 and Set 14 is shown in Figure 4.5a and 4.5b, respectively. It is illustrated that our method can accurately

Table 4.1: Summary of the major differences between the proposed and the reference methods.

Name	Model/Data-driven	Iterative?	Reconstruction	Loss
NLR-CS	Model	Yes	Direct	Group sparsity, low rank
TVAL3	Model	Yes	Direct	ℓ_2, TV
D-AMP	Model	Yes	Direct	Denoising
ReconNet	Data	No	Direct	ℓ_2
LDAMP	Hybrid	Yes	Direct	Denoising
CSGM	Data	No	Direct	ℓ_2, Adversarial
LAPRAN	Data	No	Progressive	ℓ_2, Adversarial

reconstruct high-frequency details, such as the parallel lines, contained in the ground-truth image. In contrast, the reference methods produce noticeable artifacts and start to lose details at the CR of 20.



(a) CS reconstruction results at the CR of 5.

(b) CS reconstruction results at the CR of 20.

Figure 4.5: Visual comparison of butterfly (Set 5) and zebra (Set14) at the CRs of 5 and 20. LAPRAN better preserves details.

Table 4.2: Quantitative evaluation of state-of-the-art CS reconstruction methods.

Algorithm	Compression	MNIST	CIFAR10	Set5	Set14
	ratio (CR)	SSIM/PSNR	SSIM/PSNR	SSIM/PSNR	SSIM/PSNR
NLR-CS	5	0.408/24.85	0.868/37.91	0.803/30.42	0.794/29.42
D-AMP		0.983/37.78	0.968/41.35	0.852/33.74	0.813/31.17
TVAL-3		0.934/36.39	0.847/32.03	0.812/31.54	0.727/29.48
ReconNet		0.911/29.03	0.871/32.55	0.824/31.78	0.763/29.70
CSGM		0.748/28.94	0.788/30.34	0.619/27.31	0.575/26.18
LDAMP		0.797/31.93	0.971/41.54	0.866/32.26	0.781/30.07
LAPRAN (ours)		0.993/38.46	0.978/42.39	0.895/34.79	0.834/32.71
NLR-CS	10	0.416/21.98	0.840/33.39	0.764/28.89	0.716/27.47
D-AMP		0.963/35.51	0.822/30.78	0.743/27.72	0.649/25.84
TVAL-3		0.715/27.18	0.746/29.21	0.702/28.29	0.615/26.65
ReconNet		0.868/28.98	0.843/29.78	0.779/29.53	0.704/27.45
CSGM		0.589/27.49	0.784/29.83	0.560/25.82	0.514/24.94
LDAMP		0.446/22.40	0.899/34.56	0.796/29.46	0.687/27.70
LAPRAN (ours)		0.990/38.38	0.943/38.13	0.849/32.53	0.775/30.45
NLR-CS	20	0.497/21.79	0.820/31.27	0.729/26.73	0.621/24.88
D-AMP		0.806/28.56	0.402/16.86	0.413/16.72	0.329/15.12
TVAL-3		0.494/21.00	0.623/25.77	0.583/25.18	0.513/24.19
ReconNet		0.898/27.92	0.806/29.08	0.731/27.07	0.623/25.38
CSGM		0.512/27.54	0.751/30.50	0.526/25.04	0.484/24.42
LDAMP		0.346/17.01	0.756/28.66	0.689/27.00	0.591/24.48
LAPRAN (ours)		0.985/37.02	0.896/34.12	0.801/30.08	0.716/28.39
NLR-CS	30	0.339/17.47	0.703/27.26	0.580/22.93	0.581/22.93
D-AMP		0.655/21.47	0.183/10.62	0.230/10.88	0.136/9.31
TVAL-3		0.381/18.17	0.560/24.01	0.536/24.04	0.471/23.20
ReconNet		0.892/25.46	0.777/29.32	0.623/25.60	0.598/24.59
CSGM		0.661/27.47	0.730/27.73	0.524/24.92	0.464/23.97
LDAMP		0.290/15.03	0.632/25.57	0.572/24.75	0.510/22.74
LAPRAN (ours)		0.962/31.28	0.840/31.47	0.693/28.61	0.668/27.09

4.4.4 Reconstruction Speed

We compare the runtime of each reconstruction method for reconstructing a 64×64 image patch to benchmark reconstruction speed. For the optimization-based methods, GPU acceleration is impractical due to their iterative natures. Thus, we use an Intel(R) Xeon E5-2695 CPU to run the codes provided by the respective authors. For the DNN-based methods, we use an Nvidia GTX TitanX GPU to accelerate the reconstruction process. The average runtime for each method is shown in Table 4.3. The proposed LAPRAN taks about 6ms to reconstruct 64×64 image patch, which is 4 orders of magnitude faster than NLR-CS and TVAL3, and 2 orders of magnitude faster than BM3D-AMP and LDAMP. Although our method is slightly slower than ReconNet and CSGM, it is sufficiently fast for performing real-time reconstruction.

Table 4.3: Runtime (seconds) for reconstructing a 64×64 image patch. Unlike the model-based methods, the runtime of LAPRAN is invariant to CR. LAPRAN is slightly slower than ReconNet and CSGM because of its large model capacity. LDAMP is relatively slower due to its iterative nature.

Name	Device	CR=5	CR=10	CR=20	CR=30
NLR-CS	CPU	1.869e1	1.867e1	1.833e1	1.822e1
TVAL3	CPU	1.858e1	1.839e1	1.801e1	1.792e1
BM3D-AMP	CPU	4.880e-1	4.213e-1	3.018e-1	2.409e-1
ReconNet	GPU	2.005e-3	1.703e-3	1.524e-3	1.661e-3
CSGM	GPU	1.704e-3	1.562e-3	1.490e-3	1.481e-3
LDAMP	GPU	3.556e-1	2.600e-1	1.998e-1	1.784e-1
LAPRAN	GPU	6.241e-3	6.384e-3	6.417e-3	6.008e-3

4.5 Conclusions

In this chapter, we present a scalable LAPRAN for high-fidelity, flexible, and fast CS image reconstruction. The LAPRAN consists of multiple stages of RANs that progressively reconstruct an image in multiple hierarchies. At each pyramid level, CS measurements are fused with a low-dimensional contextual latent vector to generate a high-frequency image residual, which is subsequently upsampled via a transposed CNN. The generated image residual is then added to a low-frequency image upscaled from the output of the previous level to form the final output of the current level with both higher resolution and reconstruction quality. The hierarchical nature of the LAPRAN is the key to enabling high-fidelity CS reconstruction with a flexible resolution that can be adaptive to a wide range of CRs. Each RAN in the LAPRAN can be trained independently with weight transfer to achieve faster convergence and improved accuracy. The use of contextual input at each stage and the divide-and-conquer strategy in training are the keys to achieving excellent reconstruction performance.

Chapter 5

DEEP LEARNING FOR VIDEO COMPRESSIVE SENSING RECONSTRUCTION

5.1 Introduction

High-frame-rate cameras are capable of capturing videos at frame rates over 100 frames per second (fps). These devices were originally developed for research purposes, e.g., to characterize events that occur at a rate that traditional cameras are incapable of recording in physical and biological science. Some high-frame-rate cameras, such as Photron SA1, SA3, are capable of recording high resolution still images of ephemeral events such as a supersonic flying bullet or an exploding balloon with negligible motion blur and image distortion artifacts. However, due to the complex sensor hardware designed for high sampling frequency, these types of equipment are extremely expensive (over tens of thousand dollars for one camera). The high cost limits the field of their applications. Furthermore, the high transmission bandwidth and the large storage space associated with the high frame rate challenges the manufacture of affordable consumer devices. For example, true high-definition-resolution (1080p) video cameras at a frame rate of 10k fps can generate about 500 GB data per second, which imposes significant challenges on existing transmission and storage techniques. Also, the high throughput raises energy efficiency a big concern. For example, “GoPro 5” can capture videos at 120 fps with 1080p resolution. However, the short battery life (1-2 hours) has significantly narrowed their practical applications.

Traditional video encoder, e.g., H.264/MPEG-4, is composed of motion estimation, frequency transform, quantization, and entropy coding modules. From both speed and cost perspectives, the complicated structure makes these video encoder unsuitable for high-frame-rate video cameras. Alternatively, compressive sensing (CS) is a much more hardware-friendly acquisition technique that allows video capture with a sub-Nyquist sampling rate. The advent of CS has led to the emergence of new image devices, e.g., single-pixel cameras Duarte *et al.* (2008). CS has also been applied in many practical applications, e.g., accelerating magnetic resonance imaging (MRI) Ma *et al.* (2008). While traditional signal acquisition methods follow a sample-then-compress procedure, CS could perform compression along with sampling. The novel acquisition strategy has enabled low-cost on-sensor data compression, relieving the pain for high transmission bandwidth and large storage space. In the recent decade, many algorithms have been proposed Candès *et al.* (2006a); Needell and Tropp (2010); Beck and Teboulle (2009); Daubechies *et al.* (2010); Tropp and Gilbert (2007); Blumensath and Davies (2009) to solve the CS reconstruction problem. Generally, these reconstruction algorithms are based on either optimization or greedy approaches using signal sparsity as prior knowledge. As a result, they all suffer from high computational complexity, which requires seconds to minutes to recover an image depending on the resolution. Therefore, these sparsity-based methods cannot satisfy the real-time decoding need of high-frame-rate cameras, and they are not appropriate for the high-frame-rate video CS application.

The slow reconstruction speed of conventional CS approaches motivates us to directly model the inverse mapping from the compressed domain to original domain, which is shown in Figure 5.1. Usually, this mapping is extremely complicated and difficult to model. However, the existence of massive unlabeled video data gives a chance to learn such a mapping using data-driven methods. In this work, we design an enhanced recurrent convolutional neural network (RCNN) to solve this problem. RCNN has shown astonishingly good performance for video recognition and descrip-

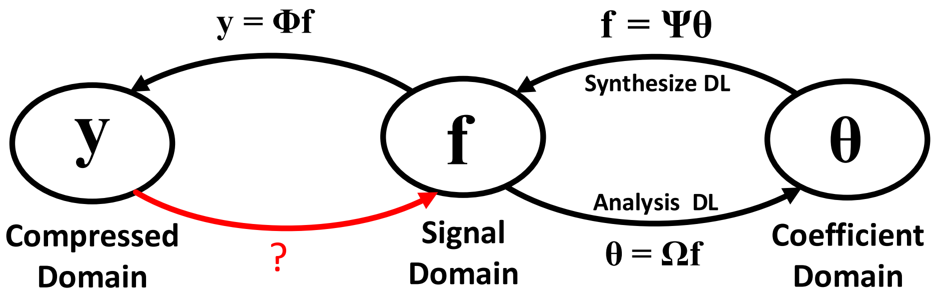


Figure 5.1: Illustration of domain transformations in CS. This work bridges the gap between compressed and signal domains.

tion Donahue *et al.* (2017); Venugopalan *et al.* (2015); Xu *et al.* (2015); Srivastava *et al.* (2015). However, conventional RCNNs are not well suited for video CS application, since they are mostly designed to extract discriminant features for classification related tasks. Simultaneously improving compression ratio (CR) and preserving visual details for high-fidelity reconstruction is a more challenging task. To solve this problem, we develop a special RCNN, called "CSVideoNet", to extract spatial-temporal features, including background, object details, and motions, to significantly improve the compression ratio and recovery quality trade-off for video CS application over existing approaches.

The contributions of this paper are summarized as follows:

- We propose an end-to-end and data-driven framework for video CS. The proposed network directly learns the inverse mapping from the compressed videos to the original input without additional pre/post-processing. To the best of our knowledge, there has been no published work that addresses this problem using similar methods.

- We propose a multi-level compression strategy to improve CR with the preservation of high-quality spatial resolution. Besides, we perform implicit motion estimation to improve temporal resolution. By combining both spatial and temporal features, we further improve the compression ratio and recovery quality trade-off without increasing much computational complexity.

- We demonstrate CSVideoNet outperforms the reference approaches not only in recovery quality but also in reconstruction speed because of its non-iterative nature. It enables real-time high-fidelity reconstruction for high-frame-rate videos at high CRs. We achieve state-of-the-art performance on the large-scale video dataset UCF-101. Specifically, CSVideoNet reconstructs videos at 125 fps on a Titan X GPU and achieves 25dB PSNR at a 100x CR.

5.2 Related Work

There have been many recovery algorithms proposed for CS reconstruction, which can be categorized as follows:

Conventional Model-based CS Recovery: In Sankaranarayanan *et al.* (2013), the authors model the evolution of scenes as a linear dynamical system (LDS). This model comprises two sub-models: the first is an observation model that models frames of video lying on a low-dimensional subspace; the second predicts the smoothly varied trajectory. The model performs well in stationary scenes, however, inadequate for non-stationary scenes.

In Yang *et al.* (2014), the authors use Gaussian mixture model (GMM) to recover high-frame-rate videos, and the reconstruction can be efficiently computed as an analytical solution. The hallmark of the algorithm is that it adapts temporal compression rate based upon the complexity of the scene. The parameters in GMM are trained off-line and tuned during the recovery process.

In Sankaranarayanan *et al.* (2015), the authors propose a multi-scale video recovery framework. It first obtains a low-resolution video preview with very low computational complexity, and then it exploits motion estimates to recover the full-resolution video by solving an optimization problem. In a similar work Fowler *et al.* (2012), the authors propose a motion-compensated and block-based CS reconstruction algorithm with smooth projected Landweber (MC-BCS-SPL). The motion vector is estimated from a reference and a reconstructed frame. The reconstructed video is derived from the combination of the low-resolution video and the estimated motion vector. The drawback of the two work is the requirement of specifying the resolution at which the preview frame is recovered, which requires prior knowledge of the object speed. Also, the recovery performance is highly dependent on the quality of motion estimation. To accurately estimate motion vector is a challenging task especially in high-frame-rate scenarios. The high computational cost further makes this model inadequate for reconstructing high-frame-rate videos.

Deep Neural Network (DNN) Based CS Recovery: In Mousavi *et al.* (2015), the authors propose a stacked autoencoder to learn a representation of the training data and to recover test data from their sub-sampled measurements. Compared to the conventional iterative approaches, which usually need hundreds of iterations to converge, the feed-forward deep neural network runs much faster in the inference stage.

In Kulkarni *et al.* (2016), the authors propose a convolutional neural network, which takes CS measurements of an image as input and outputs an intermediate reconstruction. The intermediate output is fed into an off-the-shelf denoiser to obtain the final reconstructed image. The author shows the network is highly robust to sensor noise and can recover visually higher quality images than competitive algorithms at low CRs of 10 and 25. Both Mousavi *et al.* (2015) and Kulkarni *et al.* (2016) are designed for image reconstruction, which only focus on spatial feature extraction. For video applications, temporal features between adjacent frames are also important. Therefore, the overlook of temporal correlation makes the image reconstruction algorithms inadequate for video applications.

In Iliadis *et al.* (2018), the authors propose a Video CS reconstruction algorithm based on a fully-connected neural network. This work focuses on temporal CS where multiplexing occurs across the time dimension. A 3D volume is reconstructed from 2D measurements by a feed-forward process. The author claims the reconstruction time for each frame can be reduced to about one second. The major drawback of this work is that the algorithm is based on a plain fully-connected neural network, which is not efficient in extracting temporal features.

5.3 Methodology

5.3.1 Overview of the Proposed Framework for Video CS

Two kinds of CS cameras are being used today. Spatial multiplexing cameras (SMC) take significantly fewer measurements than the number of pixels in the scene to be recovered. SMC has low spatial resolution and seeks to spatially super-resolve videos. In contrast, temporal multiplexing cameras (TMC) have a high spatial resolution but low frame-rate sensors. Due to the missing of inter frames, extra computation is needed for motion estimation. For these two sensing systems, either spatial or temporal resolution is sacrificed for achieving a better spatial-temporal trade-off. To solve this problem, we propose a new sensing and reconstruction framework, which combines the advantage of the two systems. The random video measurements are collected by SMC with very high temporal resolution. To compensate for the low spatial resolution problem in SMC, we propose a multi-CR strategy. The first *key frame* in a group of pictures (GOP) is compressed with a low CR, and the remaining *non-key frames* are compressed with a high CR. The spatial features in the key frame are reused for the recovery of the entire GOP due to the high inter-frame correlation in high-frame-rate videos. The spatial resolution is hence improved. The RNN extrapolates motion from high-resolution frames and uses it to improve the temporal resolution. Therefore, a better compression ratio and spatial-temporal resolution trade-off are obtained by the proposed framework.

The overall architecture of the proposed video CS reconstruction framework is shown in Figure 5.2. The network contains three modules: 1) an encoder (sensing matrix) for simultaneous sampling and compression; 2) a dedicated CNN for spatial features extraction after each compressed frame; 3) an LSTM for motion estimation and video reconstruction. As mentioned earlier, to improve the spatial resolution, the random encoder encodes the key frame in a GOP with more measurements and the remaining with less. Also, our previous study Xu *et al.* (2017) shows that sensing matrix can be trained with raw data to better preserve the Restricted Isometry Property (RIP). Therefore, the encoder can also be integrated into the entire model and trained with the whole network to improve reconstruction performance. Besides, as the proposed algorithm eliminates the sparsity prior constraint, the direct optimization of RIP preservation in Xu *et al.* (2017) is not necessary. Instead, we can use the reconstruction loss to train the sensing matrix along with the model. For simplicity, we still use a random Bernoulli matrix for information encoding in the experiment. Different from the prior work that extracts motion from low-resolution previews, the proposed LSTM network infers motion from high-resolution frames generated by multi-rate CNNs. The resolution of the reconstructed video is further improved with the incorporation of high-quality motion estimation.

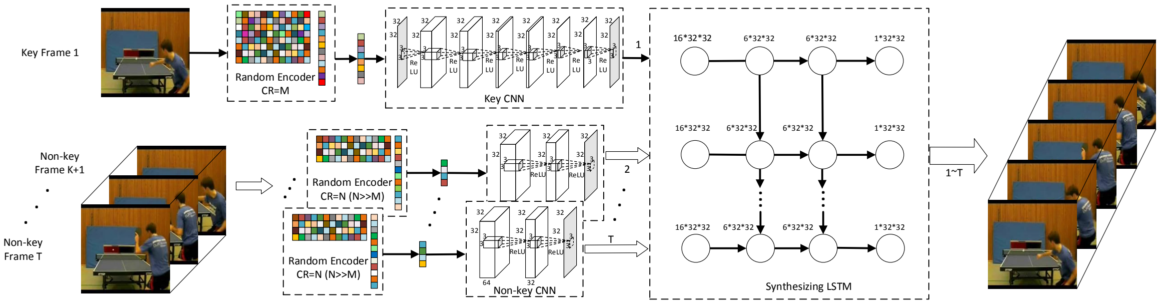


Figure 5.2: Overall architecture of the proposed framework. The compressed video frames are acquired by compressive sensing. In a length T GOP, the first one frame and the remaining (T-1) frames are compressed with a low and high CR, respectively. The reconstruction is performed by the CSVideoNet that is composed of a key CNN, multiple non-key CNNs, and a synthesizing LSTM.

Multi-rate CNN Encoder for Compression Ratio Enhancement

Typical CNN architectures used for recognition, classification, and segmentation that map input to rich hierarchical visual features is not applicable to the reconstruction problem. The goal of the CNN is not only to extract spatial visual features but also to preserve details as much as possible. Therefore, we eliminated the pooling layer which causes information loss. Also, we discard the convolution-deconvolution architecture (widely used in segmentation tasks Noh *et al.* (2015)), which first encodes salient visual features into low-dimension space and then interpolates the missing information to generate a high-resolution image. Instead, we design a special CNN suitable for CS reconstruction, which has the best recovery performance among all the tested structures mentioned above. The overall network structure is shown in Figure 5.3. All feature maps have the same dimension as the reconstructed video frames, and the number of feature maps decreases monotonically. This process resembles the sparse coding stage in CS, where a subset of dictionary atoms is combined to form the estimation of the original input. There is a fully-connected (FC) layer, denoted in gray color in Figure 5.3, which converts vectorized m-dimensional video data to 2D features maps. To reduce the latency of the system and to simplify the network architecture, we use video blocks as input and set the block size n to 32×32. All convolutional layers are followed by a ReLU layer except the final layer. We pre-train an eight-layer *key CNN* to process the key frame that is compressed with a low CR. For other non-key frames compressed with a high CR, we use 3-layer *non-key CNNs* to handle them since they carry information of low entropy. All weights of the non-key CNNs are shared to reduce the requirement of storage. Hence the proposed framework can be easily generalized to other high-frame-rate video applications that require a larger number of non-key frames. It should be noted that the pre-training of the key CNN is critical for improving the reconstruction performance. In the case where the whole network is trained from scratch without any pre-training, the convergence performance is bad. The reason is partly due to the vanishing gradients, since we have a long path from the CNNs to the LSTM. The pre-training greatly alleviate this problem.

Motion-estimation Synthesizing LSTM Decoder for Spatial-temporal Resolution Enhancement

The proposed framework is end-to-end trainable, computationally efficient, and requires no pre/post-processing. This is achieved by performing motion estimation implicitly, which is different from prior works Sankaranarayanan *et al.* (2015); Yang *et al.* (2014); Fowler *et al.* (2012). We utilize an LSTM network to extract motion features that are critical for improving temporal resolution from the CNN output. Since the information flows from the first LSTM node to the remaining, the LSTM will implicitly infers representations for the hidden motion from the key frame to the non-key frames. Therefore, the recovery quality of the GOP is improved by the aggregation of motion and spatial visual features. That is why we call this network the *motion-estimation synthesizing LSTM*. For simplicity, each input LSTM node in the experiment accepts input data with equal length. In fact, since the non-key frames carry less information than the key frame, the LSTM network can be designed to accept inputs with variable lengths. Hence, we can further reduce the model size and get a faster reconstruction speed. From the experiment results, we find the utilization of the LSTM network is critical to improving recovery fidelity. As a result, our model outperforms the competitive algorithms by a significant margin.

The update of the LSTM units is as follows:

$$\begin{aligned}
\mathbf{i}_t &= \sigma\left(W_{xi}\mathbf{x}_t + W_{hi}\mathbf{h}_{t-1} + W_{ci}\mathbf{c}_{t-1} + \mathbf{b}_i\right), \\
\mathbf{f}_t &= \sigma\left(W_{xf}\mathbf{x}_t + W_{hf}\mathbf{h}_{t-1} + W_{cf}\mathbf{c}_{t-1} + \mathbf{b}_f\right), \\
\mathbf{c}_t &= \mathbf{f}_t\mathbf{c}_{t-1} + \mathbf{i}_t \tanh\left(W_{xc}\mathbf{x}_t + W_{hc}\mathbf{h}_{t-1} + \mathbf{b}_c\right), \\
\mathbf{o}_t &= \sigma\left(W_{xo}\mathbf{x}_t + W_{ho}\mathbf{h}_{t-1} + W_{co}\mathbf{c}_t + \mathbf{b}_o\right), \\
\mathbf{h}_t &= \mathbf{o}_t \tanh(\mathbf{c}_t),
\end{aligned}$$

where $\mathbf{x}_t$ is the visual feature output of the CNN encoder. The detailed information flow and the output dimension at each LSTM node is shown in Figure 5.2. The number on the LSTM nodes denotes the dimension of the output features. Specifically, the output feature map of each CNN has a dimension of 16x32x32. All these feature maps are directly fed into the input nodes of the LSTM. The LSTM has two hidden layers, the dimension of the output of each hidden layer is 6x32x32. The dimension of the final output is 1x32x32.

5.3.2 Learning Algorithm

Given the ground-truth video frames $x_{\{1,\cdots,T\}}$ and the corresponding compressed frames $y_{\{1,\cdots,T\}}$, we use mean square error (MSE) as the loss function, which is defined as:

$$L(\mathbf{W}, \mathbf{b}) = \frac{1}{2N}\sum_{i}^{T} \|f(y_i; \mathbf{W}, \mathbf{b}) - x_i\|_2^2, \tag{5.1}$$

where $\mathbf{W}$, $\mathbf{b}$ are network weights and biases, respectively.

Using MSE as the loss function favors high PSNR. PSNR is a commonly used metric to quantitatively evaluate recovery quality. From the experiment results, we illustrate that PSNR is partially correlated to the perceptual quality. To derive a better perceptual similarity metric will be a future work. The proposed framework can be easily adapted to a new loss function.

Three training algorithms, i.e., SGD, Adagrad Duchi *et al.* (2011) and Adam Kingma and Ba (2015) are compared in the experiment. Although consuming most GPU memory, Adam converges towards the best reconstruction results. Therefore, Adam is chosen to optimize the proposed network.

5.4 Experiment

As there is no standard dataset designed for video CS, we use UCF-101 dataset introduced in Soomro *et al.* (2012) to benchmark the proposed framework. This dataset consists of 13k clips and 27 hours of video recording data collected from YouTube, which belong to 101 action classes. Videos in the dataset are randomly split into 80% for training, 10% for validation and the remaining for testing. Videos in the dataset have a resolution of 320×240 and are sampled at 25 fps. We retain only the luminance component of the extracted frames and crop the central 160×160 patch from each frame. These patches are then segmented into 32×32 non-overlapping image blocks. We get 499,760 GOPs for training and testing in total.

We set three test cases with CRs of 25, 50 and 100, respectively. Since the CR for key and non-key frames are different in the proposed method, we derive and define the CR for a particular GOP as follows. Let $m1, m2$ denotes the dimension of compressed key and non-key frame, respectively. Let n denotes the dimension of raw frames. T

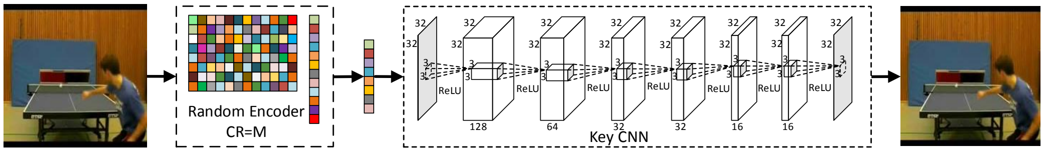


Figure 5.3: Pre-training of the key CNN.

is the sequential length of a GOP.

$$\begin{aligned} CR_1 =& n/m1, CR_2 = n/m2, \\ CR =& \frac{CR_1 \times 1 + CR_2 \times (T-1)}{T}. \end{aligned} \tag{5.2}$$

In the experiment, the CR of each key frame is m1=5, and the CR of non-key frames in each test case is m2=27, 55, and 110, respectively. Therefore, the averaged CR for each test case is about 25, 50, and 100, respectively.

The dimension of data for pre-training the key CNN is $(N \times C \times H \times W)$, where N=100 is the batch size, C=1 is the channel size, and W, H=(32, 32) is the height and width of each image block, respectively. The dimension of the data used for training the entire model is $(N' \times T \times C \times H \times W)$, where T=10 is the sequence length for one GOP, and N'=20 is the batch size. The other dimensions are the same. We shrink the batch size here because of the GPU memory limitation. In every ten consecutive video frames, we define the first one as the key frame, and the remaining as non-key frames.

5.4.1 Comparison with the State-of-the-art

We compare our algorithm with six reference work for CS reconstruction: Yang *et al.* (2014); Fowler *et al.* (2012); Mousavi *et al.* (2015); Metzler *et al.* (2016); Kulkarni *et al.* (2016); Iliadis *et al.* (2018). We summarize all baseline approaches and our approach in Table 5.1. For a fair comparison, we also re-train reference algorithms using UCF-101 dataset. Three metrics: Peak signal-to-noise ratio (PSNR), structural

Table 5.1: Summary of major differences between the proposed approach and all baselines.

Image CS	Iterative Based	Denoising-based approximate message passing	D-AMP Metzler *et al.* (2016)
	Non-iterative Based	Stacked denoising autoencoder	SDA Mousavi *et al.* (2015)
		Convolutional neural network	ReconNet Kulkarni *et al.* (2016)
Video CS	Iterative Based	Motion-compensated block-based CS with smooth projected Landweber	MC-BCS-SPL Fowler *et al.* (2012)
		Gaussian mixture model	GMM Yang *et al.* (2014)
	Non-iterative Based	Fully-connected neural network	VCSNet Iliadis *et al.* (2018)
		Proposed approach	**CSVideoNet**

similarity (SSIM) Wang *et al.* (2004), and pixel-wise mean absolute error (MAE) are applied for performance evaluation. Note that MAE is the averaged absolute error of each pixel value within the range of [0,255], which gives a straightforward measure of the pixel-wise distortion. The authors of VCSNet only offer a pre-trained model with CR of 16, without providing sufficient training details to reproduce the experiment at present. Therefore, we train the proposed model and compare it with CVSNet at a single CR of 16.

Comparison with Image CS Approaches

We first compare with the algorithms used for image CS reconstruction. D-AMP is a representative of the conventional iterative algorithms developed for CS, e.g., matching pursuit, orthogonal mating pursuit, iterative hard-thresholding. It offers state-of-the-art recovery performance and operates tens of times faster compared to other iterative methods Metzler *et al.* (2016). Both SDA and ReconNet are DNN-based reconstruction approaches for images proposed recently. Specifically, ReconNet is based on CNN and achieves state-of-the-art performance among all image CS reconstruction algorithms Kulkarni *et al.* (2016). In the experiment, we tested both frame-based and block-based D-AMP that reconstructs an entire frame and an image block at a time, respectively. For other approaches, we test them in a block-based pattern to reduce the difficulty for training the models. The quantized results of average PSNR, SSIM, and MAE for each method under different CRs are shown in Table 5.2. It is shown that CSVideoNet outperforms the reference approaches on all three metrics by a meaningful margin, especially at the CR of 100. The MAE of CSVideoNet is 4.59 at a 100x CR which means the averaged pixel-wise distortion is only $4.59/255 = 1.2\%$ compared to the ground-truth video. The PSNR drop from the CR of 25 to 100 is also calculated in Table 5.2. We found the proposed approach suffers from the least performance degradation. This is partly due to the feature sharing between the key and non-key frames when the compressed input carries limited information.

For visual quality assessment purpose, we list the reconstructed frame by each approach in Figure 5.4. The reconstructed frame is the middle (fifth) frame in a GOP. We find all the reconstructed non-key frames have homogeneous recovery quality, and the key frame has slightly better reconstruction quality than the non-key frames. As the proportion of key and non-key frames is 1:9, and the reconstruction quality of the video is dominated by that of the non-key frames. Therefore, the middle frame (a non-key frame) shown in Figure 5.4 well represents the average reconstruction quality.

For all the numerical results, we calculate all the quality metrics, including PSNR, SSIM, and MAE, by averaging the results over all frames in a GOP. We can see that CSVideoNet provides the finest details among all approaches. The edges produced by CSVideoNet is much sharper, while such details are no longer preserved by other methods after reconstruction. This comparison demonstrates that the temporal correlation is critical for video reconstruction, the overlook of such features will signifi-

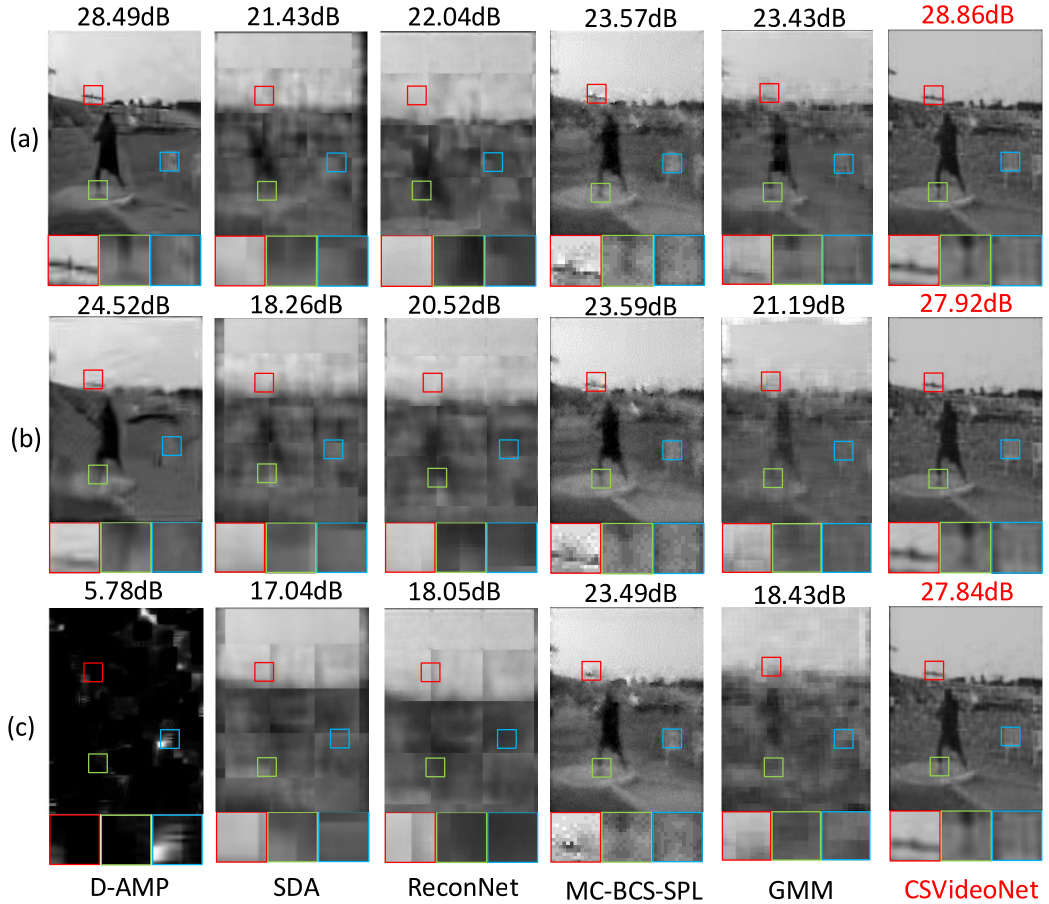


Figure 5.4: Illustration of reconstruction results for each method at the CR of (a) 25, (b) 50, and (c) 100, respectively.

cantly degrade the recovery quality of videos. Therefore, the conventional image CS approaches are not suitable for video applications.

5.4.2 *Comparison with Video CS Approaches*

We compare the proposed CSVideoNet with existing video CS approaches. MC-BCS-SPL estimates motion directly from the current and the reference frame. GMM models the spatial-temporal correlation by assuming all pixels within a video patch are drawn from a GMM distribution. GMM has the state-of-the-art performance among conventional model-based video CS approaches Yang *et al.* (2014). To the best of our knowledge, Iliadis *et al.* (2018) is the only DNN-based work proposed for

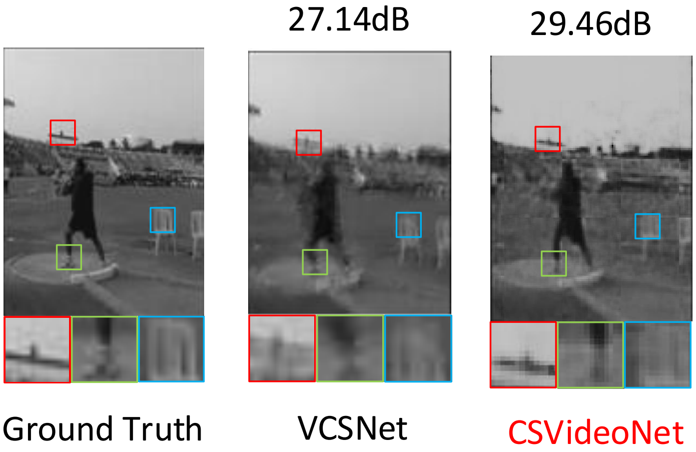


Figure 5.5: Illustration of reconstruction results at the CR of 16.

Table 5.2: Performance comparison with image CS reconstruction approaches.

	CR	D-AMP(F)	D-AMP(B)	SDA	ReconNet	CSVideoNet
PSNR	25	25.34	15.1494	23.39	24.27	**26.87**
	50	12.49	9.1719	21.96	22.47	**25.09**
	100	7.17	8.0942	20.40	20.44	**24.23**
SSIM	25	0.76	0.0934	0.69	0.73	**0.81**
	50	0.08	0.0249	0.65	0.67	**0.77**
	100	0.03	0.0067	0.61	0.61	**0.74**
MAE	25	4.65	24.92	5.76	5.02	**3.38**
	50	64.30	81.67	6.60	5.67	**4.31**
	100	92.12	86.04	8.50	7.42	**4.59**
PSNR↓	25 → 100	72%	13%	47%	16%	**10%**

video CS. The quantized results of average PSNR, SSIM, and MAE for each method under different CRs are shown in Table 5.3. It is observed that the proposed approach improves PSNR by 3 to 5dB over the reference methods. Specifically, we find MC-BCS-SPL and GMM have similar performance and perform much better than the model-based image CS approach, D-AMP. However, their performance are similar to SDA and ReconNet, which are designed for processing images. This implies that the conventional model-based methods suffer from limited performance due to the limited model capacity when dealing with large-scale problem. Even though they consider the

Table 5.3: Performance comparison with video CS reconstruction approaches.

	CR	MC-BCS-SPL	GMM	CSVideoNet
PSNR	25	22.41	23.76	**26.87**
	50	20.59	21.26	**25.09**
	100	19.67	19.64	**24.23**
SSIM	25	0.37	0.72	**0.81**
	50	0.30	0.61	**0.77**
	100	0.19	0.54	**0.74**
MAE	25	11.88	5.14	**3.38**
	50	16.03	7.50	**4.31**
	100	28.86	9.37	**4.59**
PSNR↓	25 → 100	26%	17%	**10%**

temporal correlation among video frames, the model capacity is insufficient for visual patterns. To improve performance, one could increase the size of the conventional models. However, the computational complexity forof these meods will also increase substantially, inhibiting their application to video CS.

DNN provides a viable solution. Both CSVideoNet and VCSNet are designed for video CS reconstruction. For reasons explained earlier, we compare the two approaches at a CR of 16. The results are shown in Table 5.4 and Figure 5.5. Both the two approaches achieve high recovery quality compared to other baselines. However, VCSNet is a plain fully-connect network that has limited capability for processing sequential data. As a result, it suffers from a low-quality motion estimation, which explains why it has inferior performance compared to the proposed solution.

To illustrate that the performance improvement of the proposed approach comes from integrating temporal features through the LSTM network rather than simply increasing the model size, we set another experiment, in which we compare the per-

Table 5.4: Performance comparison with VCSNet at the CR of 16.

	VCSNet	CSVideoNet
PSNR	25.07704	28.078
SSIM	0.817669	0.8431
MAE	3.887867	2.9452

Table 5.5: Structures of CNN1 and CNN2.

# Layer	1	2	3	4	5	6	7	8	9	10	11	12	13
CNN1	1	128	64	32	32	16	16	1					
CNN2	1	512	256	256	128	128	64	64	32	32	16	16	1

* CNN1 is used in CSVideoNet. The dimension of all feature maps in both CNNs are 32×32.

formance of two CNNs with different sizes. The structure of the two CNNs are shown in Table 5.5, and the performance comparison is shown in Table 5.7. We can see that simply increasing the size of CNN does not provide meaningful improvement for reconstruction. This, wh be explained by the incapability of CNN to capture temporal features. The incorporation of the LSTM network improves the PSNR by up to 4 dB, which represents more than twice of error reduction. Specifically, the performance improvement increases with thealong wiachieves theits maximum wheR is 100. This explains that the implicit motion estimation by LSTM is critical to the video CS reconstruction especially at high CRs.

5.4.3 Performance under Noise

To demonstrate that the robustness of CSVideoNet to sensor noise, we conduct a reconstruction experiment with input videos contaminated by random Gaussian noise. In this experiment, the architecture of all DNN-based frameworks remains the same as in the noiseless case. We test the performance at three levels of SNR - 20dB,

Table 5.6: Runtime comparison for reconstructing a 160×160 video frame at different CRs.

Model	CR=25	CR=50	CR=100
D-AMP(F)	38.37	41.20	31.74
D-AMP(B)	8.4652	8.5498	8.4433
SDA	0.0278	0.027	0.023
ReconNet	0.064	0.063	0.061
MC-BCS	7.17	8.03	9.00
GMM	8.87	10.54	18.34
CSVideoNet	0.0094	0.0085	0.0080

Table 5.7: Performance comparison with CNN methods.

	CR	CNN1	CNN2	CSVideoNet
PSNR	25	24.27	23.74	26.87
	50	22.47	22.17	25.09
	100	20.44	20.10	24.23
SSIM	25	0.73	0.69	0.81
	50	0.67	0.65	0.77
	100	0.61	0.58	0.74
MAE	25	5.02	6.46	3.38
	50	5.67	6.23	4.31
	100	7.42	8.92	4.59

40dB, and 60dB. For each noise level, we evaluate all approaches at three CRs of 25, 50, and 100. The average PSNR achieved by each method at different CRs and noise levels are shown in Figure 5.6. It can be observed that CSVideoNet can reliably achieve a high PSNR across at different noise levels and outperform the reference methods consistently.

5.4.4 Time Complexity

We benchmark the runtime performance of different methods. Due to the iterative nature of conventional CS algorithms (D-AMP, MC-BCS-SPL, GMM), they suffer from high data-dependency and low parallelism, which is not suitable for GPU acceleration. Due to the lack of GPU solvers, we run these reference algorithms on an octa-core Intel Xeon E5-2600 CPU. Benefiting from the feedforward data-path and high data concurrency of DNN-based approaches, we accelerate CSVideoNet and other DNN-based baselines using a Nvidia GTX Titan X GPU. The time cost for fully reconstructing a video frame in the size of (160×160) are compared in Table 5.6. CSVideoNet consumes 8 milliseconds (125 fps) to reconstruct a frame at the CR of 100. This is three orders of magnitude faster than the reference methods based on iterative approaches. The time cost of VCSNet and CSVideoNet at the CR of 16 is 3.5 and 9.7 milliseconds, respectively. Through further hardware optimization, we believe CSVideoNet has the potential to be integrated into CS cameras to enable the real-time reconstruction of high-frame-rate video CS.

5.5 Conclusion

In this chapter, we present a real-time, end-to-end, and non-iterative framework for high-frame-rate video CS. A multi-rate CNN variant and a synthesizing LSTM network are developed to jointly extract spatial-temporal features. This is the key

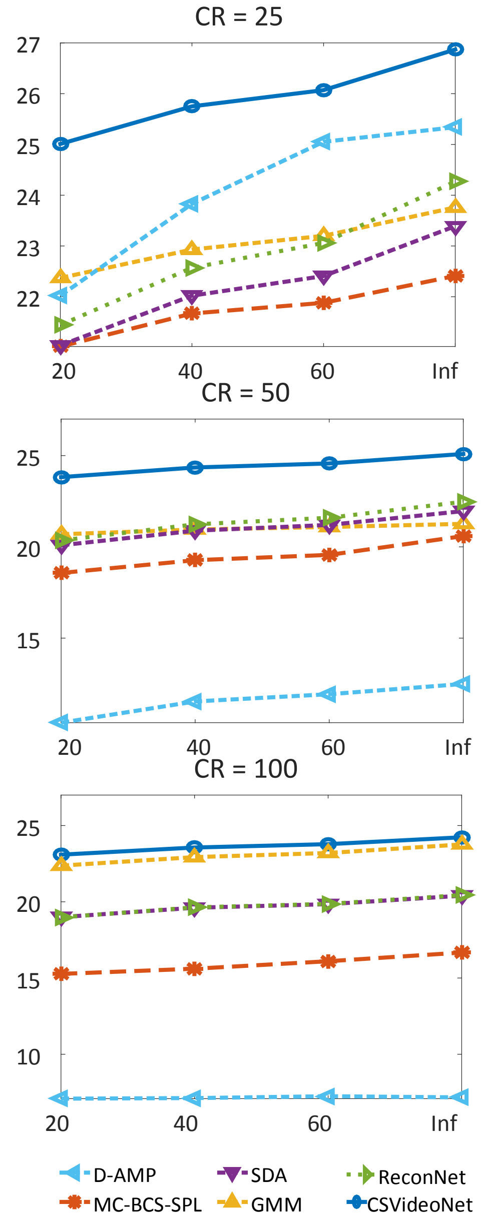


Figure 5.6: PSNR comparison at different SNRs.

to enhancing the compression ratio and recovery quality trade-off. The magnificent model capacity of the proposed deep neural network allows to map the inverse mapping of CS without exploiting any sparsity constraint. The feed-forward and high-data-concurrency natures of the proposed framework are the key to enabling GPU acceleration for real-time reconstruction. Through performance comparison, we demonstrate that CSVideoNet has the potential to be extended as a general encoding-decoding framework for high-frame-rate video CS applications. In the future work, we will exploit the effective learning methods to decode high-level information from compressed videos, e.g., object detection, action recognition, and scene segmentation.

Chapter 6

LEARNING IN THE FREQUENCY DOMAIN

6.1 Introduction

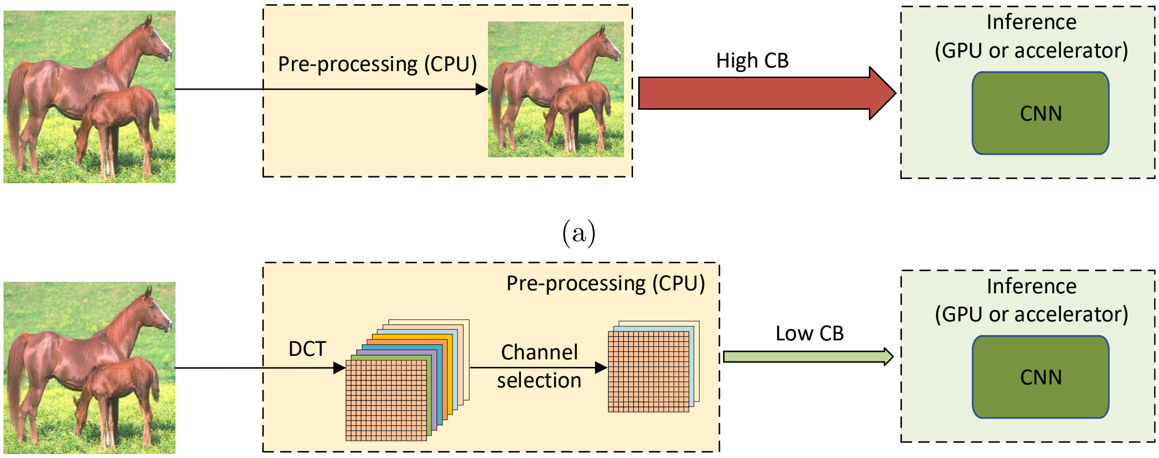
Convolutional neural networks (CNNs) have revolutionized the computer vision community because of their exceptional performance on various tasks such as image classification Krizhevsky *et al.* (2012); Karpathy *et al.* (2014), object detection Ren *et al.* (2017); Redmon *et al.* (2016), and semantic segmentation Long *et al.* (2015); Chen *et al.* (2018). Constrained by the computing resources and memory limitations, most CNN models only accept RGB images at low resolutions (*e.g.*, 224×224). However, images produced by modern cameras are usually much larger. For example, the high definition (HD) resolution images (1920×1080) are considered relatively small by modern standards. Even the average image resolution in the ImageNet dataset Russakovsky *et al.* (2015) is 482×415, which is roughly four times the size accepted by most CNN models. Therefore, a large portion of real-world images are aggressively downsized to 224×224 to meet the input requirement of classification networks. However, image downsizing inevitably incurs information loss and accuracy degradation Pei *et al.* (2019). Prior works Kim *et al.* (2018); Saeedan *et al.* (2018) aim to reduce information loss by learning task-aware downsizing networks. However, those networks are task-specific and require additional computation, which are not favorable in practical applications. In this work, we propose to reshape the high-resolution images in the frequency domain, *i.e.*, discrete cosine transform (DCT) domain [1], rather

[1] We interchangeably use the terms frequency domain and DCT domain in the context of this paper.

than resizing them in the spatial domain, and then feed the reshaped DCT coefficients to CNN models for inference. Our method requires little modification to the existing CNN models that take RGB images as input. Thus, it is a universal replacement for the routine data pre-processing pipelines. We demonstrate that our method achieves higher accuracy in image classification, object detection, and instance segmentation tasks than the conventional RGB-based methods with an equal or smaller input data size. The proposed method leads to a direct reduction in the required inter-chip communication bandwidth that is often a bottleneck in modern deep learning inference systems, *i.e.*, the computational throughput of rapidly evolving AI accelerators/GPUs is becoming increasingly higher than the data loading throughput of CPUs, as shown in Figure 6.1.

Inspired by the observation that human visual system (HVS) has unequal sensitivity to different frequency components Kim and Lee (2017), we analyze the image classification, detection and segmentation task in the frequency domain and find that CNN models are more sensitive to low-frequency channels than the high-frequency channels, which coincides with HVS. This observation is validated by a learning-based channel selection method that consists of multiple "on-off switches". The DCT coefficients with the same frequency are packed as one channel, and each switch is stacked on a specific frequency channel to either allow the entire channel to flow into the network or not.

Using the decoded high-fidelity images for model training and inference has posed significant challenges, from both data transfer and computation perspectives Wei *et al.* (2019); You *et al.* (2018). Due to the spectral bias of the CNN models, one can only keep the important frequency channels during inference without losing accuracy. In this work, we also develop a static channel selection approach to preserve the salient channels rather than using the entire frequency spectrum for inference. Experiment results show that the CNN models still retain the same accuracy when the input data size is reduced by 87.5%.



(a)

(b)

Figure 6.1: (a) The workflow of the conventional CNN-based methods using RGB images as input. (b) The workflow of the proposed method using DCT coefficients as input. CB represents the required communication bandwidth between CPU and GPU/accelerator.

The contributions of this paper are as follows:

- We propose a method of learning in the frequency domain (using DCT coefficients as input), which requires little modification to the existing CNN models that take RGB input. We validate our method on ResNet-50 and MobileNetV2 for the image classification task and Mask R-CNN for the instance segmentation task.

- We show that learning in the frequency domain better preserves image information in the pre-processing stage than the conventional spatial downsampling approach (spatially resizing the images to 224×224, the default input size of most CNN models) and consequently achieves improved accuracy, *i.e.*, +1.60% on ResNet-50 and +0.63% on MobileNetV2 for the ImageNet classification task,

+0.8% on Mask R-CNN for both object detection and instance segmentation tasks.

- We analyze the spectral bias from the frequency perspective and show that the CNN models are more sensitive to low-frequency channels than high-frequency channels, similar to the human visual system (HVS).
- We propose a learning-based dynamic channel selection method to identify the trivial frequency components for static removal during inference. Experiment results on ResNet-50 show that one can prune up to 87.5% of the frequency channels using the proposed channel selection method with no or little accuracy degradation in the ImageNet classification task.
- To the best of our knowledge, this is the first work that explores learning in the frequency domain for object detection and instance segmentation. Experiment results on Mask R-CNN show that learning in the frequency domain can achieve a 0.8% average precision improvement for the instance segmentation task on the COCO dataset.

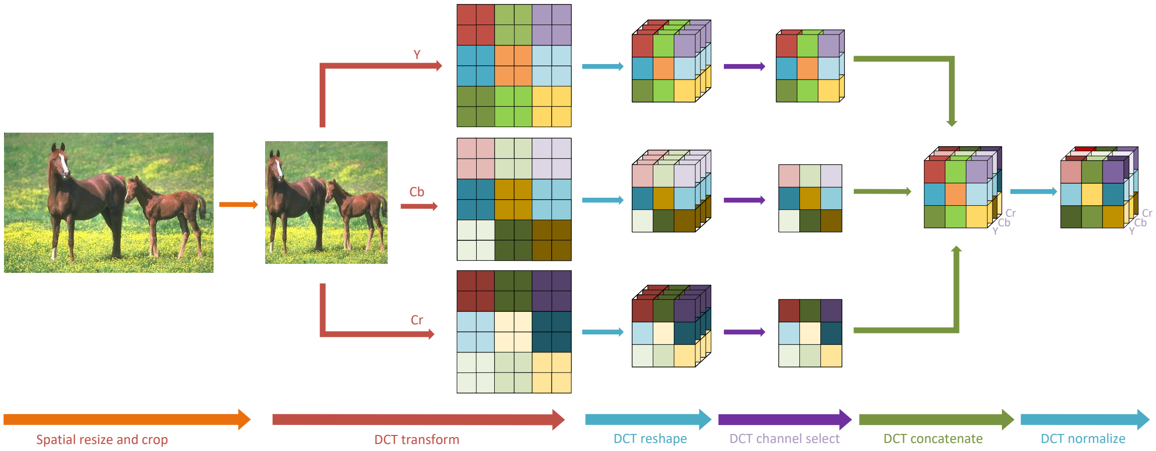


Figure 6.2: The data pre-processing pipeline for learning in the frequency domain.

6.2 Related Work

Learning in the frequency domain: Compressed representations in the frequency domain contain rich patterns for image understanding tasks. Torfason *et al.* (2018); XU *et al.* (2018b); Wu *et al.* (2018a) train dedicated autoencoder-based networks on compression and inference tasks jointly. Gueguen *et al.* (2018) extracts features from the frequency domain to classify images. Ehrlich and Davis (2019) proposes a model conversion algorithm to convert the spatial-domain CNN models to the frequency domain. Our method differs from the prior works in two aspects. First, we avoid the complex model transition procedure from the spatial to the frequency domain. Thus, our method has a broader application scope. Second, we provide an analysis method to interpret the spectral bias of neural networks in the frequency domain.

Dynamic Neural Networks: Prior works Veit and Belongie (2018); Wang *et al.* (2018); Guo *et al.* (2019); Wu *et al.* (2018b); Chen *et al.* (2019b) propose to selectively skip the convolutional blocks on the fly based on the activations of the previous blocks. These works adjust the model complexity in response to the input of each convolutional block. Only the intermediate features that are most relevant to the inputs are computed in the inference stage to reduce computation cost. In contrast, our method exclusively operates on the raw inputs and distills the salient frequency components to lower the communication bandwidth requirement for input data.

Efficient Network Training: There are substantial recent interests in training efficient networks Frankle and Carbin (2019); Molchanov *et al.* (2019); Wang *et al.* (2019); Han *et al.* (2016), which focus on network compression via kernel pruning, learned quantization, and entropy encoding. Another line of works aim to compress the CNN models in the frequency domain. Chen *et al.* (2016) reduces the storage space by converting filter weights to the frequency domain and using a hash function to group the frequency parameters into hash buckets. Wang *et al.* (2019) also transforms the kernels to the frequency domain and discards the low-energy frequency coefficients for high compression. Dziedzic *et al.* (2019) constrains the frequency spectra of CNN kernels to reduce memory consumption. These network compression works in the frequency domain all rely on the FFT-based convolution, which is generally more effective on larger kernels. Nevertheless, the state-of-the-art CNN models use small kernels, *e.g.*, 3 × 3 or 1 × 1. Extensive efforts need to be taken to optimize the computation efficiency of these FFT-based CNN models Lavin and Gray (2016). In contrast, our method makes little modification to the existing CNN models. Thus, our method requires no extra effort to improve its computation efficiency on the CNN models with small kernels. Another fundamental difference is that our method aims at reducing the input data size rather than model complexity.

6.3 Methodology

In this work, we propose a generic method on learning in the frequency domain, including a data pre-processing pipeline as well as an input data size pruning method.

Figure 6.1 shows the comparison of our method and the conventional approach. In the conventional approach, high-resolution RGB images are usually pre-processed on a CPU and transmitted to a GPU/AI accelerator for real-time inference. Because uncompressed images in the RGB format are usually large, the requirement of the communication bandwidth between a CPU and a GPU/AI accelerator is usually high. Such communication bandwidth can be the bottleneck of the system performance, as shown in Figure 6.1(a). To reduce both the computation cost and the communication bandwidth requirement, high-resolution RGB images are downsampled to smaller images, which often results in information loss and thus lower inference accuracy.

In our method, high-resolution RGB images are still pre-processed on a CPU. However, they are first transformed to the YCbCr color space and then to the frequency domain. This coincides with the most widely-used image compression standards, such as JPEG. All components of the same frequency are grouped into one channel. In this way, multiple frequency channels are generated. As shown in Section 6.3.2, certain frequency channels have bigger impact on the inference accuracy than the others. Thus, we propose to only preserve and transmit the most important frequency channels to a GPU/AI accelerator for inference. Compared to the conventional approach, the proposed method requires less communication bandwidth and achieves higher accuracy at the same time.

We demonstrate that the input features in the frequency domain can be applied to all existing CNN models developed in the spatial domain with minimal modification. Specifically, one just need to remove the input CNN layer and reserve the remaining residual blocks. The first residual layer is used as the input layer, and the number of input channels is modified to fit the dimension of the DCT coefficient inputs. As such, a modified model can maintain similar parameter count and computational complexity to the original model.

Based on our frequency-domain model, we propose a learning-based channel selection method to explore the spectral bias of a given CNN model, *i.e.*, which frequency components are more informative to the subsequent inference task. The findings motivate us to prune the trivial frequency components for inference, which significantly reduces the input data size, consequently reducing both the computational complexity of domain transformation and the required communication bandwidth, while maintaining inference accuracy.

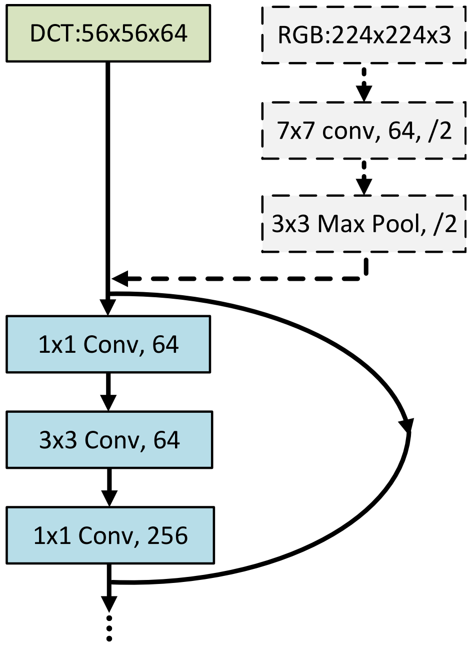


Figure 6.3: Connecting the pre-processed input features in the frequency domain to ResNet-50. The three input layers (the dashed gray blocks) in a vanilla ResNet-50 are removed to admit the 56×56×64 DCT inputs. We take 64 channels as an example. This value can vary based on the channel selection. In learning-based channel selection, all 192 channels are analyzed for their importance to accuracy, based on which only a subset ($\ll 192$ channels) is used in the static selection approach.

6.3.1 Data Pre-processing in the Frequency Domain

The data pre-processing flow is shown in Figure 6.2. We follow the pre-processing and augmentation flow in the spatial domain, consisting of image resizing, cropping, and flipping (spatial resize and crop in Figure 6.2). Then images are transformed to the YCbCr color space and converted to the frequency domain (DCT transform in Figure 6.2). The two-dimensional DCT coefficients at the same frequency are grouped into one channel to form three-dimensional DCT cubes (DCT reshape in Figure 6.2). As will be discussed in Section 6.3.2, a subset of impactful frequency channels are selected (DCT channel select in Figure 6.2). The selected channels in the YCbCr color space are concatenated together to form one tensor (DCT concatenate in Figure 6.2).

Lastly, every frequency channel is normalized by the mean and variance calculated from the training dataset.

The DCT reshape operation in Figure 6.2 groups a two-dimensional DCT coefficients to a three-dimensional DCT cube. Since the JPEG compression standard uses 8×8 DCT transformation on the YCbCr color space, we group the components of the same frequency in all the 8×8 blocks into one channel, maintaining their spatial relations at each frequency. Thus, each of the Y, Cb, and Cr components provides $8 \times 8 = 64$ channels, one for each frequency, with a total of 192 channels in the frequency domain. Suppose the shape of the original RGB input image is $H \times W \times C$, where $C = 3$ and the height and width of the image is denoted as H and W, respectively. After converting to the frequency domain, the input feature shape becomes $H/8 \times W/8 \times 64C$, which maintains the same input data size.

Since the input feature maps in the frequency domain are smaller in the H and W dimensions but larger in the C dimension than the spatial-domain counterpart, we skip the input layer of a conventional CNN model, which is usually a stride-2 convolution. If a max-pooling operator immediately follows the input convolution (*e.g.*, ResNet-50), we skip the max-pooling operator as well. Then we adjust the channel size of the next layer to match the number of channels in the frequency domain. It is illustrated in Figure 6.3. This way, we minimally adjust the existing CNN models to accept the frequency-domain features as input.

In the image classification task, the CNN models usually take input features of the shape $224 \times 224 \times 3$, which is usually downsampled from images with a much higher resolution. When the classification is performed in the frequency domain, larger images can be taken as input. Take ResNet-50 as an example, the input features in the frequency domain are connected to the first residue block with the number of channels adjusted to 192, forming an input feature of the shape $56 \times 56 \times 192$, as shown in Figure 6.2. That is DCT-transformed from input images of size $448 \times 448 \times 3$, which preserves four times more information than the $224 \times 224 \times 3$ counterpart in the spatial domain, at the cost of 4 times the input feature size. Similarly, for the model MobileNetV2, the input feature shape is $112 \times 112 \times 192$, reshaped from images of size $896 \times 896 \times 3$. As discussed in Section 6.3.3, the majority of the frequency channels can be pruned without sacrificing accuracy. The frequency channel pruning operation is referred to as DCT channel select in Figure 6.2.

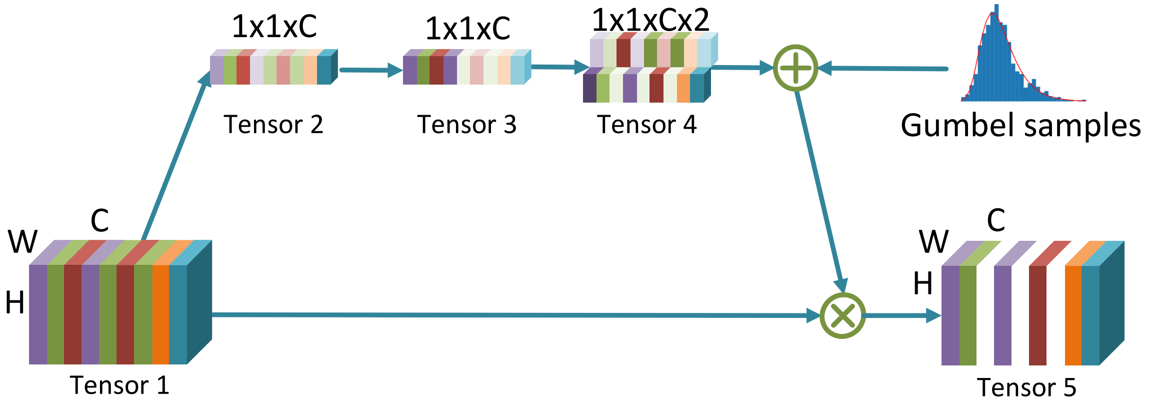


Figure 6.4: The gate module that generates the binary decisions based on the features extracted by the SE-Block. The white color channels of Tensor 5 indicate the unselected channels.

6.3.2 Learning-based Frequency Channel Selection

As different channels of the input feature are at different frequencies, we conjecture that some frequency channels are less informative to the subsequent tasks such as image classification, object detection, and instance segmentation, and removing the trivial frequency channels shall not result in performance degradation. Thus, we propose a learning-based channel selection mechanism to exploit the relative importance of each input frequency channel. We employ a dynamic gate module that assigns a binary score to each frequency channel. The salient channels are rated as one, the others as zero. The input frequency channels with zero scores are detached from the network. Thus, the input data size is reduced, leading to reduced computation complexity of domain transformation and communication bandwidth requirement. The proposed gate module is simple and can be part of the model to be applied in online inference.

Figure 6.4 describes our proposed gate module in detail. The input is of shape $W \times H \times C$ ($C = 192$ in this chapter), with C frequency channels (Tensor 1 in Figure 6.4). It is first converted to Tensor 2 in Figure 6.4 of shape $1 \times 1 \times C$ by average pooling. Then it is converted to Tensor 3 in Figure 6.4 of shape $1 \times 1 \times C$ by a 1×1 convolutional layer. Conversion from Tensor 1 to Tensor 3 is exactly the same as a two-layer squeeze-and-excitation block (SE-Block) Hu *et al.* (2018), which utilizes the channel-wise information to emphasize the informative features and suppress the trivial ones. Then, Tensor 3 is converted to Tensor 4 in Figure 6.4 of the shape $1 \times 1 \times C \times 2$ by multiplying every element in Tensor 3 with two trainable parameters. During inference, the two numbers for each of the 192 channels in Tensor 4 are normalized and serve as the probability of being sampled as 0 or 1, and then, point-wise multiplied to the input frequency channels to obtain Tensor 5 in Figure 6.4. As an example, if the two numbers in the ith channel in Tensor 4 are 7.5 and 2.5, there is a 75% probability that the ith gate is turned off. In other words, the ith frequency channel in Tensor 5 becomes all zeros 75% of the times, which effectively blocks this frequency channel from being used for inference.

Our gate module differs from the conventional SE-Block in two ways. First, the proposed gate module outputs a tensor of dimension $1 \times 1 \times C \times 2$, where the two numbers in the last dimension describe the probability of being on and off for each frequency channel, respectively. Thus we add another 1×1 convolution layer for the conversion. Second, the number multiplied to each frequency channel is either 0 or 1, *i.e.*, a binary decision of using the frequency or not. The decision is obtained by sampling a Bernoulli distribution Bern(p), where p is calculated by the 2 numbers in the $1 \times 1 \times C \times 2$ tensor mentioned above.

One of the challenges in the proposed gate module is that the Bernoulli sampling process is not differentiable in case one needs to update the weights in the gate module. Jang *et al.* (2017); Tucker *et al.* (2017); Maddison *et al.* (2017) propose a reparameterization method, called Gumbel Softmax trick, which allows the gradients to back propagate through a discrete sampling process (see Gumbel samples in Figure 6.4).

Let $\boldsymbol{x} = (x_1, x_2, \ldots, x_C)$ be the input channels in the frequency domain ($C = 192$) for a CNN model. Let $\mathbf{F}$ denote the proposed gate module such that $\mathbf{F}(x_i) \in \{0, 1\}$, for each frequency channel x_i. Then x_i is selected if

$$\mathbf{F}(x_i) \neq 0, \text{ i.e., } \mathbf{F}(x_i) \odot x_i \neq \mathbf{0}, \tag{6.1}$$

where $\odot$ is the element-wise product.

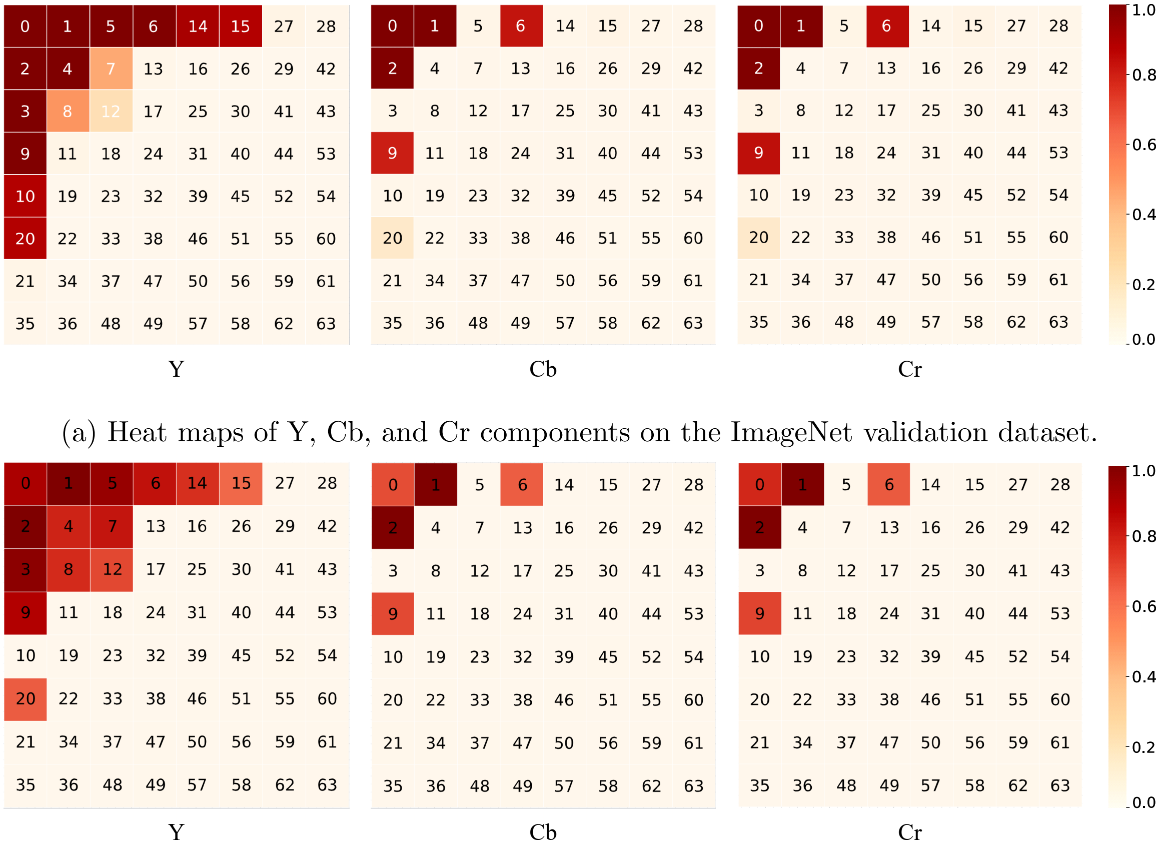
We add a regularization term to the loss function that balances the number of selected frequency channels, which is minimized together with the cross-entropy loss or other accuracy-related loss. Our loss function is thus as follows,

$$\mathcal{L} = \mathcal{L}_{Acc} + \lambda \cdot \sum_{i=1}^{C} \mathbf{F}(x_i), \tag{6.2}$$

where $\mathcal{L}_{Acc}$ is the loss that is related to accuracy. λ is a hyperparameter indicating the relative weight of the regularization term.

6.3.3 Static Frequency Channel Selection

The learning-based channel selection provides a dynamic estimation of the importance of each frequency channel, *i.e.*, different input images may have different subsets of the frequency channels activated.



(a) Heat maps of Y, Cb, and Cr components on the ImageNet validation dataset.

(b) Heat maps of Y, Cb, and Cr components on the COCO validation dataset

Figure 6.5: A heat map visualization of input frequency channels on the ImageNet validation dataset for image classification and COCO validation dataset for instance segmentation. The numbers in each square represent the corresponding channel indices. The color from bright to dark indicates the possibility of a channel being selected from low to high.

To understand the pattern of frequency channel activation, we plot two heat maps, one on the classification task (Figure 6.5a) and one on the segmentation task (Figure 6.5b). The number in each box indicates the frequency index of the channel, with a lower and higher index indicating a lower and higher frequency, respectively. The heat map value indicates the likelihood a frequency channel being selected for inference across all the validation images.

Based on the patterns in the heat maps shown in Figure 6.5, we make several observations:

- The low-frequency channels (boxes with small indices) are selected much more often than the high-frequency channels (boxes with with large indices). This demonstrates that low-frequency channels are more informative than the high-frequency channels in general for vision inference tasks.
- The frequency channels in luma component Y are selected more often than the channels in chroma components Cb and Cr. This indicates that the luma component is more informative for vision inference tasks.
- The heat maps share a common pattern between the classification and segmentation tasks. This indicates that the above-mentioned two observations are not specific to one task and is very likely to be general to more high-level vision tasks.
- Interestingly, some lower frequency channels have lower probability of being selected than the slightly higher frequency channels. For example, in Cb and Cr components, both tasks favor Channel 6 and 9 over Channel 5 and 3.

Those observations imply that the CNN models may indeed exhibit similar characteristics to the HVS, and the image compression standards (*e.g.*, JPEG) targeting human eyes may be suitable for the CNN models as well.

The JPEG compression standard puts more bits to the low-frequency and the luma components. Following the same principle, we statically select the lower frequency channels, with more emphasis on the luma component than the chroma components. This ensures the frequency channels with higher activation probabilities are fed into the CNN models. The rest of the frequency channels can be pruned by either the image encoder or decoder to reduce the required data transmission bandwidth and input data size.

Table 6.1: ResNet-50 classification results on ImageNet (validation). The input size of each method is normalized over the baseline ResNet-50. The input frequency channels are selected with the square and triangle channel selection pattern if the postfix S and T is specified, respectively.

ResNet-50	#Channels	Size Per Channel	Top-1	Top-5	Normalized Input Size
RGB	3	224×224	75.780	92.650	1.0
YCbCr	3	224×224	75.234	92.544	1.0
DCT-192 Gueguen *et al.* (2018)	192	28×28	76.060	93.020	1.0
DCT-192 (ours)	192	56×56	77.194	93.454	4.0
DCT-24D (ours)	24	56×56	77.166	93.560	0.5
DCT-24S (ours)	24	56×56	77.196	93.504	0.5
DCT-24T (ours)	24	56×56	77.148	93.326	0.5
DCT-48S (ours)	48	56×56	77.384	93.554	1.0
DCT-48T (ours)	48	56×56	77.338	93.614	1.0
DCT-64S (ours)	64	56×56	77.232	93.624	1.3
DCT-64T (ours)	64	56×56	77.280	93.456	1.3

Table 6.2: MobileNetV2 classification results on ImageNet (validation).

MobileNetV2	#Channels	Size Per Channel	Top-1	Top-5	Normalized Input Size
RGB	3	224×224	71.702	90.415	1.0
DCT-6S (ours)	6	112×112	71.776	90.258	0.5
DCT-12S (ours)	12	112×112	72.156	90.634	1.0
DCT-24S (ours)	24	112×112	72.364	90.606	2.0
DCT-32S (ours)	32	112×112	72.282	90.592	2.7

6.4 Experiment Results

We benchmark our proposed methodology on three different high-level vision tasks: image classification, detection, and segmentation.

6.4.1 *Experiment Settings on Image Classification*

We benchmark our method on image classification using the ImageNet 2012 Large-Scale Visual Recognition Challenge dataset (ILSVRC-2012) Deng *et al.* (2009). We use the stochastic gradient descent (SGD) optimizer. SGD is applied with an initial learning rate of 0.1, a momentum of 0.9, and a weight decay of 4e-5. We choose ResNet-50 He *et al.* (2016) and MobileNetV2 Sandler *et al.* (2018) as the CNN models because they contain important building blocks (*e.g.*, residue blocks and depthwise separable convolutions) widely used in modern CNN models. Note that our method can be generally applied to any CNN model. We train 210 and 150 epochs and decay the learning rate by 0.1 every 50 epochs for ResNet-50 and MobileNetV2, respectively.

To normalize the input channels, we compute the mean and variance of the DCT coefficients for each of the 192 frequency channels separately on all the training images.

As described in Section 6.3.1, the input features in the frequency domain are generated from images with a much higher resolution than the spatial-domain counterpart. However, some of the images in the ImageNet dataset have lower resolutions. We perform similar pre-processing steps as in the spatial domain, including resizing and cropping to a larger image size, performing upsampling when needed.

6.4.2 Experiment Results on Image Classification

We train the ResNet-50 model with 192 frequency channel inputs on the image classification task using the approach described in Section 6.3.2. The gate module for channel selection is trained together with the ResNet-50 model. Figure 6.5a shows a heat map of the selection results over the validation set with $\lambda = 0.1$. Note that different regularization parameters λ generate different number of activated frequency channels in heat maps. A typical example is shown in Figure 6.5a, that most channels ($\geq 80\%$) have very low possibility ($\leq 3\%$) of being selected.

Observing that low frequency channels are more important in the heat maps, we explore the sensitivity of the precise shapes of selected channels. In Table 6.1, DCT-

24D shows the accuracy when 24 (14+5+5) channels are precisely selected based on the result of the dynamic selection in Figure 6.5a. In comparison, DCT-24T and DCT-24S show the accuracy when a total of 24 channels for Y, Cb, Cr components are close to upper-left triangles and squares, respectively. The variation of the top-1 accuracy is almost negligible and all of them outperform a baseline ResNet-50 by roughly 1.4%. This demonstrates that the benefit of the proposed frequency-domain learning can be applied to many tasks as long as a majority of low-frequency channels are selected. Note the input data size is only a half of the baseline ResNet-50. Since DCT-24S provides a slightly better result, the remaining static selection are based on patterns that are close to upper-left squares (some lower right channels may be missing).

Similarly, we choose the top $(32, 8, 8)$ channels for DCT-48S/T and top $(44, 10, 10)$ channels for DCT-64S/T. The results on the ImageNet dataset are shown in Table 6.1 along with selecting all 192 frequency channels. In particular, compared with the baseline ResNet-50, the top-1 accuracy is improved by 1.4% using all frequency channels. One should also note that the accuracy is dropped when the inputs are transformed from the RGB to the YCbCr color space (both in the spatial domain) by roughly 0.5%, and the improvement of our method (in the frequency domain) over the YCbCr case is even larger.

Another interesting observation is that the model trained with a subset of channels may perform better than the model trained with all the 192 channels. Such a counter-intuitive observation implies that a small number (e.g., 24) of low-frequency channels are sufficient to capture useful features and additional frequency components may introduce noise.

Similar experiments are performed using the MobileNetV2 as the baseline CNN model and the results are shown in Table 6.2. Note that DCT-12S and DCT-6S select

12 and 6 frequency channels, and the input data size is the same and a half of the baseline MobileNetV2, respectively. The top-1 accuracy of DCT-12S and DCT-6S is improved by 0.454% and 0.074%, respectively. The top-1 accuracy is improved by 0.662% and 0.580% by selecting 32 and 24 frequency channels, respectively.

Table 6.3: Bbox AP results of Mask R-CNN using different backbones on COCO 2017 validation set. The baseline Mask R-CNN uses a ResNet-50-FPN as the backbone. The DCT method uses the frequency-domain ResNet-50-FPN as the backbone.

Backbone	#Channels	Size Per Channel	bbox					
			AP	AP@0.5	AP@0.75	AP$_S$	AP$_M$	AP$_L$
ResNet-50-FPN (RGB)	3	800×1333	37.3	59.0	40.2	21.9	40.9	48.1
DCT-24S (ours)	24	200×334	37.7	59.2	40.9	21.7	41.4	49.1
DCT-48S (ours)	48	200×334	38.1	59.5	41.2	22.0	41.3	49.8
DCT-64S (ours)	64	200×334	38.1	59.6	41.1	22.5	41.6	49.7

Table 6.4: Mask AP results of Mask R-CNN using different backbones on COCO 2017 validation set.

Backbone	#Channels	Size Per Channel	mask					
			AP	AP@0.5	AP@0.75	AP$_S$	AP$_M$	AP$_L$
ResNet-50-FPN (RGB)	3	800×1333	34.2	55.9	36.2	15.8	36.9	50.1
DCT-24S (ours)	24	200×334	34.6	56.1	36.9	16.1	37.4	50.7
DCT-48S (ours)	48	200×334	35.0	56.6	37.2	16.3	37.5	52.3
DCT-64S (ours)	64	200×334	35.0	56.5	37.4	16.9	37.6	51.6

6.4.3 Experiment Settings on Instance Segmentation

We train our model on the COCO train2017 split containing about 118k images and evaluate on the val2017 split containing 5k images. We evaluate the bounding box (bbox) average precision (AP) for the object detection task and the mask AP for the instance segmentation task. Based on the Mask R-CNN He *et al.* (2017), our model consists of a frequency-domain ResNet-50 model as introduced in Section 6.4.1 and a feature pyramid network Lin *et al.* (2017) as the backbone. The frequency-domain

ResNet-50 model is fine-tuned with the bounding-box recognition head and the mask prediction head. Input images are resized to a maximum scale of 1600×2666 without changing the aspect ratio. The corresponding DCT coefficients have a maximum size of 200×334, which are fed into the ResNet-50-FPN Lin *et al.* (2017) for feature extraction.

We train our networks for 20 epochs with an initial learning rate of 0.0025, which is decreased by 10× after 16 and 19 epochs. The rest of the configurations follow those of MMDetection Chen *et al.* (2019a).

In Table 6.3 and Table 6.4, we report the AP metric that averages APs across IoU thresholds from 0.5 to 0.95 with an interval of 0.05. Both the bbox AP and the mask AP are evaluated. For the mask AP, we also report AP@0.5 and AP@0.75 at the IoU threshold of 0.5 and 0.75 respectively, as well as AP_S, AP_M, and AP_L at different scales.

6.4.4 Experiment Results on Instance Segmentation

We train our Mask R-CNN model using the 192-channel inputs in the frequency domain for instance segmentation. The gate module for dynamic channel selection is trained together with the entire Mask R-CNN. Figure 6.5b shows the heat maps for the dynamic selection.

We further train our models using only the top 24, 48, and 64 high-probability frequency channels. The bbox and mask AP of our method in different cases is reported in Table 6.3 and Table 6.4, respectively. The experiment results show that our method outperforms the RGB-based Mask R-CNN baseline with both an equal (DCT-48S) or smaller (DCT-24S) input data size. Specifically, the 24-channel model (DCT-24S) achieves an improvement of 0.4 in both bbox AP and mask AP with a half of the input data size compared to the RGB-based Mask R-CNN baseline.

Figure 6.6 visually illustrates the segmentation results of the Mask R-CNN model trained and performing inference in the frequency domain.

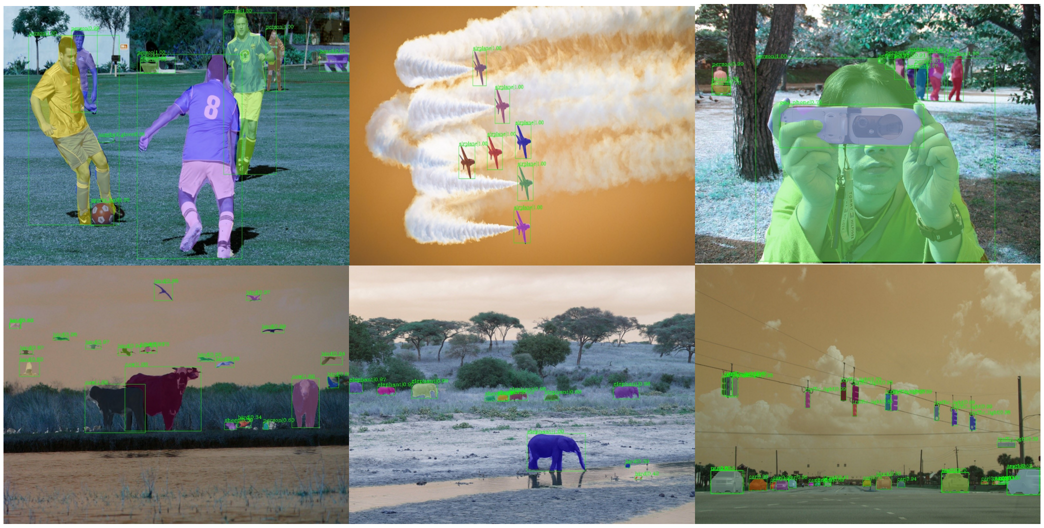


Figure 6.6: Examples of instance segmentation results on the COCO dataset.

6.5 Conclusion

In this chapter, we propose a method of learning in the frequency domain and demonstrate its generality and superiority for a variety of tasks, including classification, detection, and segmentation. Our method requires little modification to the existing CNN models that take RGB input thus can be generally applied to existing network training and inference methods. We show that the proposed method better preserves image information in the pre-processing stage than the conventional spatial downsampling approach and consequently achieves improved accuracy. We propose a learning-based dynamic channel selection method and empirically show that the CNN models are more sensitive to low-frequency channels than high-frequency channels. Experiment results show that one can prune up to 87.5% of the frequency channels using the proposed channel selection method with no or little accuracy degradation in the classification, object detection, and instance segmentation tasks.

Chapter 7

CONCLUSION

In this dissertation, we have systematically introduced our work on compressed reconstruction (CR) and compressed learning (CL). We demonstrate the effort on building CR and CL systems from the data-driven perspective.

We propose a number of models for the CR of bio-signals, images, and videos. Specifically, a scalable Laplacian pyramid reconstructive adversarial network (LAPRAN) is proposed for single-image compressed reconstruction, which progressively reconstructs images following the concept of Laplacian pyramid. LAPRAN provides high-fidelity recovery quality with a flexible resolution that is adaptive to a wide range of compression ratios. For the CR of videos, we propose CSVideoNet that is composed of a multi-rate CNN and a synthesizing RNN to improve the trade-off between compression ratio (CR) and spatial-temporal resolution of the reconstructed videos.

We also propose a CR framework that can directly extract features from the compressed data for image classification, objection detection, and semantic/instance segmentation. We provide an algorithm for analyzing the spectral bias of neural network from the frequency perspective, and propose a learning-based frequency selection method to identify the trivial frequency components which can be removed without accuracy loss. Compared with the conventional spatial downsampling approaches, our model learning in the frequency domain can achieve higher accuracy with reduced input data size.

Although this field is rapidly progressing within the past few decade, many critical questions are still open.

1. The integration of data compression and model compression. Though data compression and model compression are considered as two independent task, the two tasks might be highly correlated since they both aim to reduce data dimension. The co-optimization of model and data may be beneficial for both tasks and may improve the performance of neural networks according to the information bottleneck theory Tishby and Zaslavsky (2015).

2. Compressed reconstruction (CR) meets few-shot learning. Most existing data-driven CR frameworks are still data-intensive and thus prevent their application scenarios. Few-Shot Learning can potentially expand the flexibility of CR systems by rapidly generalizing to new tasks containing only a few samples available.

3. Compressed learning (CL) beyond Fourier-related transforms. It is still unknown which transformation will deliver better compression-accuracy trade-off than the Fourier-related transforms (such as DFT and DCT). Following the concept of learning image/video compression, a learned network-based compression model may be treated as the desired transform.

4. Frequency-based neural network architecture design. It is still unknown how the performance of CNN impacted by the frequency response of each CNN kernel. Frequency analysis may provide an alternative perspective for explaining neural networks and the network architecture design may be influenced accordingly.

5. How to apply CL on 3D data such as point clouds. The vast amount of generated data by 3D sensing devices opens a new direction to develop 3D CL systems that can effectively compress 3D data and inference directly on the compressed

data. Autonomous driving, augmented reality (AR), virtual reality (VR) and mixed reality (MR) applications may benefit from the 3D CL system.

6. CR meets reinforcement learning (RL). The deterministic sensing matrix used for sampling in CR delivers sub-optimal recovery quality. A better sampling policy may be by found out by leveraging RL through searching in a vast sampling space.

REFERENCES

"Gartner identifies top 10 strategic iot technologies and trends", https://www.gartner.com/en/newsroom/press-releases/2018-11-07-gartner-identifies-top-10-strategic-iot-technologies-and-trends (2018).

Abo-Zahhad, M., A. Hussein and A. Mohamed, "Compression of ecg signal based on compressive sensing and the extraction of significant features", Journal of Communications, Network and System Sciences **8**, 97–117 (2015).

Aharon, M., M. Elad and A. Bruckstein, "K-svd: An algorithm for designing overcomplete dictionaries for sparse representation", IEEE Transactions on Signal Processing **54**, 11, 4311–4322 (2006).

Ansari-Ram, F. and S. Hosseini-Khayat, "Ecg signal compression using compressed sensing with nonuniform binary matrices", in "International Symposium on Artificial Intelligence and Signal Processing", (2012).

Arbelaez, P., M. Maire, C. Fowlkes and J. Malik, "Contour detection and hierarchical image segmentation", IEEE Transactions on Pattern Analysis and Machine Intelligence **33**, 5, 898–916 (2011).

Beck, A. and M. Teboulle, "A fast iterative shrinkage-thresholding algorithm for linear inverse problems", SIAM Journal on Imaging Sciences pp. 183–202 (2009).

Becker, S., J. Bobin and E. J. Candès, "Nesta: A fast and accurate first-order method for sparse recovery", SIAM Journal on Imaging Sciences **4**, 1, 1–39 (2011a).

Becker, S. R., E. J. Candès and M. C. Grant, "Templates for convex cone problems with applications to sparse signal recovery", Mathematical Programming Computation **3**, 3, 165–218 (2011b).

Bevilacqua, M., A. Roumy, C. Guillemot and M. L. Alberi-Morel, "Low-complexity single-image super-resolution based on nonnegative neighbor embedding", in "BMVC", (2012).

Blumensath, T. and M. E. Davies, "Iterative hard thresholding for compressed sensing", Applied and Computational Harmonic Analysis **27**, 3, 265 – 274 (2009).

Bora, A., A. Jalal, E. Price and A. G. Dimakis, "Compressed sensing using generative models", in "ICML", (2017).

Bottou, L. and O. Bousquet, "The tradeoffs of large scale learning", in "NIPS", (2008).

Bruno A. Olshausen, D. J. F., "Sparse coding with an overcomplete basis set: A strategy employed by v1?", Vision Research **37**, 3311–3325 (1997).

Candès, E., J. Romberg and T. Tao, "Robust uncertainty principles: exact signal reconstruction from highly incomplete frequency information", IEEE Transactions on Information Theory **52**, 2, 489–509 (2006a).

Candès, E., J. Romberg and T. Tao, "Stable signal recovery from incomplete and inaccurate measurements", Communications on Pure and Applied Mathematics **59** (2006b).

Candès, E. J., Y. C. Eldar, D. Needell and P. Randall, "Compressed sensing with coherent and redundant dictionaries", Applied and Computational Harmonic Analysis **31**, 1, 59 – 73 (2011).

Candès, E. J. and M. B. Wakin, "An introduction to compressive sampling", IEEE Signal Processing Magazine **25**, 2, 21–30 (2008).

Candès, E. J., "The restricted isometry property and its implications for compressed sensing", Comptes Rendus Mathematique **346**, 9, 589 – 592 (2008).

Casson, A. and E. Rodriguez-Villegas, "Signal agnostic compressive sensing for body area networks: Comparison of signal reconstructions", in "Annual International Conference of the IEEE Engineering in Medicine and Biology Society", (2012).

Chae, D., Y. Alem, S. Durrani and R. Kennedy, "Performance study of compressive sampling for ecg signal compression in noisy and varying sparsity acquisition", in "ICASSP", (2013).

Chen, F., A. Chandrakasan and V. Stojanovic, "Design and analysis of a hardware-efficient compressed sensing architecture for data compression in wireless sensors", IEEE Journal of Solid-State Circuits **47**, 3, 744–756 (2012).

Chen, K., J. Wang, J. Pang, Y. Cao, Y. Xiong, X. Li, S. Sun, W. Feng, Z. Liu, J. Xu, Z. Zhang, D. Cheng, C. Zhu, T. Cheng, Q. Zhao, B. Li, X. Lu, R. Zhu, Y. Wu, J. Dai, J. Wang, J. Shi, W. Ouyang, C. C. Loy and D. Lin, "MMDetection: Open mmlab detection toolbox and benchmark", ArXiv:1906.07155 (2019a).

Chen, L.-C., Y. Zhu, G. Papandreou, F. Schroff and H. Adam, "Encoder-decoder with atrous separable convolution for semantic image segmentation", in "ECCV", (2018).

Chen, W., J. Wilson, S. Tyree, K. Q. Weinberger and Y. Chen, "Compressing convolutional neural networks in the frequency domain", in "ACM SIGKDD International Conference on Knowledge Discovery and Data Mining", (2016).

Chen, Z., Y. Li, S. Bengio and S. Si, "You look twice: Gaternet for dynamic filter selection in cnns", in "CVPR", (2019b).

Cui, Z., H. Chang, S. Shan, B. Zhong and X. Chen, "Deep network cascade for image super-resolution", in "ECCV", (2014).

Dabov, K., A. Foi, V. Katkovnik and K. Egiazarian, "Image denoising by sparse 3-d transform-domain collaborative filtering", IEEE Transactions on Image Processing **16**, 8, 2080–2095 (2007).

Daubechies, I., R. DeVore, M. Fornasier and C. S. Güntürk, “Iteratively reweighted least squares minimization for sparse recovery”, Communications on Pure and Applied Mathematics **63**, 1, 1–38 (2010).

Davenport, M. A., “Random observations on random observations: Sparse signal acquisition and processing”, ProQuest Dissertations and Theses p. 187 (2010).

Davisson, L., “Rate distortion theory: A mathematical basis for data compression”, IEEE Transactions on Communications **20**, 6, 1202–1202 (1972).

Deng, J., W. Dong, R. Socher, L.-J. Li, K. Li and L. Fei-Fei, “ImageNet: A Large-Scale Hierarchical Image Database”, in “CVPR”, (2009).

Denton, E. L., S. Chintala, a. szlam and R. Fergus, “Deep generative image models using a laplacian pyramid of adversarial networks”, in “NIPS”, (2015).

Donahue, J., L. A. Hendricks, M. Rohrbach, S. Venugopalan, S. Guadarrama, K. Saenko and T. Darrell, “Long-term recurrent convolutional networks for visual recognition and description”, IEEE Transactions on Pattern Analysis and Machine Intelligence **39**, 4, 677–691 (2017).

Dong, C., C. C. Loy, K. He and X. Tang, “Learning a deep convolutional network for image super-resolution”, in “ECCV”, (2014a).

Dong, C., C. C. Loy, K. He and X. Tang, “Image super-resolution using deep convolutional networks”, IEEE Transactions on Pattern Analysis and Machine Intelligence **38**, 2, 295–307 (2016).

Dong, W., G. Shi, X. Li, Y. Ma and F. Huang, “Compressive sensing via nonlocal low-rank regularization”, IEEE Transactions on Image Processing **23**, 8, 3618–3632 (2014b).

Duarte, M. F., M. A. Davenport, D. Takhar, J. N. Laska, T. Sun, K. F. Kelly and R. G. Baraniuk, “Single-pixel imaging via compressive sampling”, IEEE Signal Processing Magazine **25**, 2, 83–91 (2008).

Duarte-Carvajalino, J. and G. Sapiro, “Learning to sense sparse signals: Simultaneous sensing matrix and sparsifying dictionary optimization”, IEEE Transactions on Image Processing **18**, 7, 1395–1408 (2009).

Duchi, J. C., E. Hazan and Y. Singer, “Adaptive subgradient methods for online learning and stochastic optimization”, JMLR **12**, 2121–2159 (2011).

Dziedzic, A., J. Paparrizos, S. Krishnan, A. Elmore and M. Franklin, “Band-limited training and inference for convolutional neural networks”, in “ICML”, (2019).

Ehrlich, M. and L. Davis, “Deep Residual Learning in the JPEG Transform Domain”, in “ICCV”, (2019).

Elad, M., “Optimized projections for compressed sensing”, IEEE Transactions on Signal Processing **55**, 12, 5695–5702 (2007).

Elad, M. and M. Aharon, “Image denoising via sparse and redundant representations over learned dictionaries”, IEEE Transactions on Image Processing **15**, 12, 3736–3745 (2006).

Fowler, J. E., S. Mun and E. W. Tramel, “Block-based compressed sensing of images and video”, Foundations and Trends in Signal Processing **4**, 4, 297–416 (2012).

Frankle, J. and M. Carbin, “The lottery ticket hypothesis: Finding sparse, trainable neural networks”, in “ICLR”, (2019).

Glasner, D., S. Bagon and M. Irani, “Super-resolution from a single image”, in “ICCV”, (2009).

Goldberger, A. L., L. A. N. Amaral, L. Glass, J. M. Hausdorff, P. C. Ivanov, R. G. Mark, J. E. Mietus, G. B. Moody, C.-K. Peng and H. E. Stanley, “PhysioBank, PhysioToolkit, and PhysioNet: Components of a new research resource for complex physiologic signals”, Circulation **101**, 23, e215–e220 (2000).

Grant, M. and S. Boyd, “Graph implementations for nonsmooth convex programs”, in “Recent Advances in Learning and Control”, (2008).

Gueguen, L., A. Sergeev, B. Kadlec, R. Liu and J. Yosinski, “Faster neural networks straight from jpeg”, in “NIPS”, (2018).

Guo, Q., Z. Yu, Y. Wu, D. Liang, H. Qin and J. Yan, “Dynamic recursive neural network”, in “CVPR”, (2019).

Guo, W. and W. Yin, “EdgeCS: edge guided compressive sensing reconstruction”, in “Visual Communications and Image Processing”, (2010).

Han, S., H. Mao and W. J. Dally, “Deep compression: Compressing deep neural networks with pruning, trained quantization and huffman coding”, ICLR (2016).

He, K., G. Gkioxari, P. Dollár and R. Girshick, “Mask r-cnn”, in “ICCV”, (2017).

He, K., X. Zhang, S. Ren and J. Sun, “Deep residual learning for image recognition”, in “CVPR”, (2016).

He, K., X. Zhang, S. Ren and J. Sun, “Deep residual learning for image recognition”, in “CVPR”, (2016).

Hegde, C. *et al.*, “Numax: A convex approach for learning near-isometric linear embeddings”, IEEE Transactions on Signal Processing **63**, 22, 6109–6121 (2015).

Hu, J., L. Shen and G. Sun, “Squeeze-and-excitation networks”, in “CVPR”, (2018).

Huggins, P. S. and S. W. Zucker, “Greedy basis pursuit”, IEEE Transactions on Signal Processing **55**, 7, 3760–3772 (2007).

Iliadis, M., L. Spinoulas and A. K. Katsaggelos, “Deep fully-connected networks for video compressive sensing”, Digital Signal Processing **72**, 9 – 18 (2018).

Ioffe, S. and C. Szegedy, “Batch normalization: Accelerating deep network training by reducing internal covariate shift”, in “ICML”, (2015).

Jang, E., S. Gu and B. Poole, “Categorical reparameterization with gumbel-softmax”, in “ICLR”, (2017).

Karpathy, A., G. Toderici, S. Shetty, T. Leung, R. Sukthankar and L. Fei-Fei, “Large-scale video classification with convolutional neural networks”, in “CVPR”, (2014).

Kim, H., M. Choi, B. Lim and K. Mu Lee, “Task-aware image downscaling”, in “ECCV”, (2018).

Kim, J., J. K. Lee and K. M. Lee, “Accurate image super-resolution using very deep convolutional networks”, in “CVPR”, (2016).

Kim, J. and S. Lee, “Deep learning of human visual sensitivity in image quality assessment framework”, in “CVPR”, (2017).

Kingma, D. P. and J. Ba, “Adam: A Method for Stochastic Optimization”, in “ICLR”, (2015).

Krizhevsky, A., I. Sutskever and G. E. Hinton, “Imagenet classification with deep convolutional neural networks”, in “NIPS”, (2012).

Kulkarni, K., S. Lohit, P. Turaga, R. Kerviche and A. Ashok, “Reconnet: Non-iterative reconstruction of images from compressively sensed measurements”, in “CVPR”, (2016).

Lai, W.-S., J.-B. Huang, N. Ahuja and M.-H. Yang, “Deep laplacian pyramid networks for fast and accurate super-resolution”, in “CVPR”, (2017).

Lavin, A. and S. Gray, “Fast algorithms for convolutional neural networks”, in “CVPR”, (2016).

LeCun, Y., L. Bottou, G. B. Orr and K.-R. Müller, “Efficient backprop”, in “NIPS Workshop”, (1998).

Lee, H., A. Battle, R. Raina and A. Y. Ng, “Efficient sparse coding algorithms”, in “NIPS”, (2007).

Lee, S., J. Luan and P. Chou, “A new approach to compressing ecg signals with trained overcomplete dictionary”, in “Conference on Wireless Mobile Communication and Healthcare (Mobihealth)”, (2014a).

Lee, S. *et al.*, “A new approach to compressing ecg signals with trained overcomplete dictionary”, in “International Conference on Wireless Mobile Communication and Healthcare”, (2014b).

Li, C., W. Yin, and Y. Zhang, ““user’s guide for tval3: Tv minimization by augmented lagrangian and alternating direction algorithms”, in “Rice CAAM Department report”, (2009).

Lin, T., P. Dollár, R. Girshick, K. He, B. Hariharan and S. Belongie, “Feature pyramid networks for object detection”, in “CVPR”, (2017).

Long, J., E. Shelhamer and T. Darrell, “Fully convolutional networks for semantic segmentation”, in “CVPR”, (2015).

Ma, S., W. Yin, Y. Zhang and A. Chakraborty, “An efficient algorithm for compressed mr imaging using total variation and wavelets”, in “CVPR”, (2008).

Maddison, C. J., A. Mnih and Y. W. Teh, “The concrete distribution: A continuous relaxation of discrete random variables”, in “ICLR”, (2017).

Mairal, J., F. Bach, J. Ponce and G. Sapiro, “Online dictionary learning for sparse coding”, in “ICML”, (2009).

Mairal, J., F. Bach, J. Ponce and G. Sapiro, “Online learning for matrix factorization and sparse coding”, Journal of Machine Learning Research **11**, 19–60 (2010).

Mamaghanian, H., N. Khaled, D. Atienza and P. Vandergheynst, “Compressed sensing for real-time energy-efficient ecg compression on wireless body sensor nodes”, IEEE Transactions on Biomedical Engineering **58**, 9, 2456–2466 (2011).

Metzler, C., A. Mousavi and R. Baraniuk, “Learned d-amp: Principled neural network based compressive image recovery”, in “NIPS”, (2017).

Metzler, C. A., A. Maleki and R. G. Baraniuk, “From denoising to compressed sensing”, IEEE Transactions on Information Theory **62**, 9, 5117–5144 (2016).

Molchanov, P., A. Mallya, S. Tyree, I. Frosio and J. Kautz, “Importance estimation for neural network pruning”, in “CVPR”, (2019).

Mousavi, A. and R. G. Baraniuk, “Learning to invert: Signal recovery via deep convolutional networks”, in “ICASSP”, (2017).

Mousavi, A., A. B. Patel and R. G. Baraniuk, “A deep learning approach to structured signal recovery”, in “Annual Allerton Conference on Communication, Control, and Computing”, (2015).

Nam, S., M. Davies, M. Elad and R. Gribonval, “The cosparse analysis model and algorithms”, Applied and Computational Harmonic Analysis **34**, 1, 30 – 56 (2013).

Needell, D. and J. A. Tropp, “Cosamp: Iterative signal recovery from incomplete and inaccurate samples”, ACM Communications **53**, 12, 93–100 (2010).

Noh, H., S. Hong and B. Han, “Learning deconvolution network for semantic segmentation”, in “ICCV”, (2015).

Pathak, D., P. Krähenbühl, J. Donahue, T. Darrell and A. Efros, “Context encoders: Feature learning by inpainting”, in “CVPR”, (2016).

Pei, Y., Y. Huang, Q. Zou, X. Zhang and S. Wang, "Effects of image degradation and degradation removal to cnn-based image classification", IEEE Transactions on Pattern Analysis and Machine Intelligence pp. 1–1 (2019).

Polania, L., R. Carrillo, M. Blanco-Velasco and K. Barner, "Compressed sensing based method for ecg compression", in "ICASSP", (2011).

Radford, A., L. Metz and S. Chintala, "Unsupervised representation learning with deep convolutional generative adversarial networks", in "ICLR", (2015).

Redmon, J., S. Divvala, R. Girshick and A. Farhadi, "You only look once: Unified, real-time object detection", in "CVPR", (2016).

Ren, F. and D. Markovic, "18.5 a configurable 12-to-237ks/s 12.8mw sparse-approximation engine for mobile exg data aggregation", in "ISSCC", (2015).

Ren, S., K. He, R. Girshick and J. Sun, "Faster r-cnn: Towards real-time object detection with region proposal networks", IEEE Transactions on Pattern Analysis and Machine Intelligence **39**, 6, 1137–1149 (2017).

Russakovsky, O., J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, A. C. Berg and L. Fei-Fei, "ImageNet Large Scale Visual Recognition Challenge", International Journal of Computer Vision **115**, 3, 211–252 (2015).

Saeedan, F., N. Weber, M. Goesele and S. Roth, "Detail-preserving pooling in deep networks", in "CVPR", (2018).

Sandler, M., A. Howard, M. Zhu, A. Zhmoginov and L. Chen, "Mobilenetv2: Inverted residuals and linear bottlenecks", in "CVPR", (2018).

Sankaranarayanan, A., P. Turaga, R. Chellappa and R. Baraniuk, "Compressive acquisition of linear dynamical systems", SIAM Journal on Imaging Sciences **6**, 4, 2109–2133 (2013).

Sankaranarayanan, A. C., L. Xu, C. Studer, Y. Li, K. F. Kelly and R. G. Baraniuk, "Video compressive sensing for spatial multiplexing cameras using motion-flow models", SIAM Journal on Imaging Sciences **8**, 3, 1489–1518 (2015).

Schulter, S., C. Leistner and H. Bischof, "Fast and accurate image upscaling with super-resolution forests", in "CVPR", (2015).

Snoek, C. G. M., M. Worring and A. W. M. Smeulders, "Early versus late fusion in semantic video analysis", in "ACM International Conference on Multimedia", (2005).

Soomro, K., A. R. Zamir and M. Shah, "UCF101: A dataset of 101 human actions classes from videos in the wild", CoRR **abs/1212.0402** (2012).

Srivastava, N., E. Mansimov and R. Salakhudinov, "Unsupervised learning of video representations using lstms", in "ICML", (2015).

Tishby, N. and N. Zaslavsky, “Deep learning and the information bottleneck principle”, in “IEEE Information Theory Workshop”, (2015).

Torfason, R., F. Mentzer, E. Ágústsson, M. Tschannen, R. Timofte and L. V. Gool, “Towards image understanding from deep compression without decoding”, in “ICLR”, (2018).

Tropp, J. A. and A. C. Gilbert, “Signal recovery from random measurements via orthogonal matching pursuit”, IEEE Transactions on Information Theory **53**, 12, 4655–4666 (2007).

Tucker, G., A. Mnih, C. J. Maddison, J. Lawson and J. Sohl-Dickstein, “Rebar: Low-variance, unbiased gradient estimates for discrete latent variable models”, in “NIPS”, (2017).

Varshney, U., “Pervasive healthcare and wireless health monitoring”, Mobile Networks and Applications **12**, 2, 113–127 (2007).

Veit, A. and S. Belongie, “Convolutional networks with adaptive inference graphs”, in “ECCV”, (2018).

Venugopalan, S., M. Rohrbach, J. Donahue, R. Mooney, T. Darrell and K. Saenko, “Sequence to sequence - video to text”, in “ICCV”, (2015).

Wang, K., Z. Liu, Y. Lin, J. Lin and S. Han, “Haq: Hardware-aware automated quantization with mixed precision”, in “CVPR”, (2019).

Wang, X., F. Yu, Z.-Y. Dou, T. Darrell and J. E. Gonzalez, “Skipnet: Learning dynamic routing in convolutional networks”, in “ECCV”, (2018).

Wang, Y., C. Xu, C. Xu and D. Tao, “Packing convolutional neural networks in the frequency domain”, IEEE Transactions on Pattern Analysis and Machine Intelligence **41**, 10 (2019).

Wang, Y. *et al.*, “Optimizing boolean embedding matrix for compressive sensing in rram crossbar”, in “ISLPED”, (2015).

Wang, Z., A. C. Bovik, H. R. Sheikh and E. P. Simoncelli, “Image quality assessment: from error visibility to structural similarity”, IEEE Transactions on Image Processing **13**, 4, 600–612 (2004).

Wei, X., Y. Liang, P. Zhang, C. H. Yu and J. Cong, “Overcoming data transfer bottlenecks in dnn accelerators via layer-conscious memory managment”, in “ACM/SIGDA International Symposium on Field-Programmable Gate Arrays”, (2019).

Wu, C.-Y., M. Zaheer, H. Hu, R. Manmatha, A. J. Smola and P. Krähenbühl, “Compressed video action recognition”, in “CVPR”, (2018a).

Wu, Z., T. Nagarajan, A. Kumar, S. Rennie, L. S. Davis, K. Grauman and R. Feris, “Blockdrop: Dynamic inference paths in residual networks”, in “CVPR”, (2018b).

Xu, K., J. Ba, R. Kiros, K. Cho, A. Courville, R. Salakhudinov, R. Zemel and Y. Bengio, “Show, attend and tell: Neural image caption generation with visual attention”, in “ICML”, (2015).

Xu, K., Y. Li and F. Ren, “An energy-efficient compressive sensing framework incorporating online dictionary learning for long-term wireless health monitoring”, in “ICASSP”, (2016).

Xu, K., Y. Li and F. Ren, “A data-driven compressive sensing framework tailored for energy-efficient wearable sensing”, in “ICASSP”, (2017).

Xu, K., M. Qin, F. Sun, Y. Wang, Y.-K. Chen and F. Ren, “Learning in the frequency domain”, in “CVPR”, (2020).

Xu, K. and F. Ren, “Csvideonet: A real-time end-to-end learning framework for high-frame-rate video compressive sensing”, in “WACV”, (2018).

XU, K., Z. Zhang and F. Ren, “Lapran: A scalable laplacian pyramid reconstructive adversarial network for flexible compressive sensing reconstruction”, in “ECCV”, (2018a).

XU, K., Z. Zhang and F. Ren, “Lapran: A scalable laplacian pyramid reconstructive adversarial network for flexible compressive sensing reconstruction”, in “ECCV”, (2018b).

Yang, J., J. Wright, T. S. Huang and Y. Ma, “Image super-resolution via sparse representation”, IEEE Transactions on Image Processing **19**, 11, 2861–2873 (2010).

Yang, J., X. Yuan, X. Liao, P. Llull, D. J. Brady, G. Sapiro and L. Carin, “Video compressive sensing using gaussian mixture models”, IEEE Transactions on Image Processing **23**, 11, 4863–4878 (2014).

You, Y., Z. Zhang, C.-J. Hsieh, J. Demmel and K. Keutzer, “Imagenet training in minutes”, in “International Conference on Parallel Processing”, (2018).

Zeiler, M. D. and R. Fergus, “Visualizing and understanding convolutional networks”, in “ECCV”, (2014).

Zeiler, M. D., G. W. Taylor and R. Fergus, “Adaptive deconvolutional networks for mid and high level feature learning”, in “ICCV”, (2011).

Zeyde, R., M. Elad and M. Protter, “On single image scale-up using sparse-representations”, in “Curves and Surfaces”, (2012).

Zigel, Y., A. Cohen and A. Katz, “The weighted diagnostic distortion (wdd) measure for ecg signals compression”, IEEE Transactions on Biomedical Engineering **47**, 11, 1422–1430 (2000).

APPENDIX A

APPENDIX TO LAPRAN

Reconstructed Images

We have shown the visual comparison on Set 5 and Set 14 at the CR of 5 and 20 in Figure5 and Figure6, respective. Besides, more reconstructed images on Set 5 and Set 14 at the CR of 5, 10, 20 and 30 are shown in FigureA.1, A.2, A.3, and A.4, respectively.

Network Architecture

The network architecture of RecGen1, RecDisc1, RecGen2, RecDisc2, RecGen3, RecDisc3, RecGen4 and RecDisc4 is shown in Table A.1, A.2, A.3, and A.4, respectively.

Table A.1: Network structure of the first stage of LAPRAN.

Layer Name	Output Size	Kernel	Stride	Pad
Input	3x51			
Reshape	153			
Linear	4096			
Reshape	64x8x8			
conv1	64x8x8	3,3	1,1	1,1
bn1	64x8x8			
Resblk	64x8x8			
conv2	3x8x8	3,3	1,1	1,1
tanh	3x8x8			

(a) Network structure of RecGen1.

Layer Name	Output Size	Kernel	Stride	Pad
Input	3x8x8			
conv1	32x8x8	3,3	1,1	1,1
bn1	32x8x8			
conv2	32x4x4	3,3	2,2	1,1
bn2	32x4x4			
conv3	64x4x4	3,3	1,1	1,1
bn3	64x4x4			
conv4	1	4,4	1,1	0,0

(b) Network structure of RecDisc1.

Table A.2: Network structure of the second stage of LAPRAN.

Layer Name	Output Size	Kernel	Stride	Pad
Input	3x8x8			
conv1	64x8x8	3,3	1,1	1,1
bn1	64x8x8			
conv2	64x4x4	3,3	2,2	1,1
bn2	64x4x4			
Reshape1	1024			
Linear1	306			
Fuse	612			
Linear2	4096			
Reshape2	64x8x8			
Deconv1	64x16x16	4,4	2,2	1,1
Resblk	64x16x16			
conv3	3x16x16	3,3	1,1	1,1
tanh	3x16x16			

(a) Network structure of RecGen2.

Layer Name	Output Size	Kernel	Stride	Pad
Input	3x16x16			
conv1	32x16x16	3,3	1,1	1,1
bn1	32x16x16			
conv2	32x8x8	3,3	2,2	1,1
bn2	32x8x8			
conv3	64x8x8	3,3	1,1	1,1
bn3	64x8x8			
conv4	64x4x4	3,3	2,2	1,1
bn4	64x4x4			
conv5	128x4x4	3,3	1,1	1,1
bn5	128x4x4			
conv6	1	4,4	1,1	0,0

(b) Network structure of RecDisc2.

Table A.3: Network structure of the third stage of LAPRAN.

Layer Name	Output Size	Kernel	Stride	Pad
Input	3x16x16			
conv1	64x16x16	3,3	1,1	1,1
bn1	64x16x16			
conv2	64x8x8	3,3	2,2	1,1
bn2	64x8x8			
Reshape1	4096			
Linear1	612			
Fuse	1224			
Linear2	16384			
Reshape2	64x16x16			
Deconv1	64x32x32	4,4	2,2	1,1
Resblk	64x32x32			
conv3	3x32x32	3,3	1,1	1,1
tanh	3x32x32			

(a) Network structure of RecGen3.

Layer Name	Output Size	Kernel	Stride	Pad
Input	3x32x32			
conv1	32x32x32	3,3	1,1	1,1
bn1	32x32x32			
conv2	32x16x16	3,3	2,2	1,1
bn2	32x16x16			
conv3	64x16x16	3,3	1,1	1,1
bn3	64x16x16			
conv4	64x8x8	3,3	2,2	1,1
bn4	64x8x8			
conv5	128x8x8	3,3	1,1	1,1
bn5	128x8x8			
conv6	128x4x4	3,3	2,2	1,1
bn6	128x4x4			
conv7	256x4x4	3,3	1,1	1,1
bn7	256x4x4			
conv8	1	4,4	1,1	0,0

(b) Network structure of RecDisc3.

Table A.4: Network structure of the fourth stage of LAPRAN.

Layer Name	Output Size	Kernel	Stride	Pad
Input	3x32x32			
conv1	64x32x32	3,3	1,1	1,1
bn1	64x32x32			
conv2	64x16x16	3,3	2,2	1,1
bn2	64x16x16			
Reshape1	16384			
Linear1	1227			
Fuse	2354			
Linear2	65536			
Reshape2	64x32x32			
Deconv1	64x64x64	4,4	2,2	1,1
Resblk	64x64x64			
conv3	3x64x64	3,3	1,1	1,1
tanh	3x64x64			

(a) Network structure of RecGen4.

Layer Name	Output Size	Kernel	Stride	Pad
Input	3x64x64			
conv1	32x64x64	3,3	1,1	1,1
bn1	32x64x64			
conv2	32x32x32	3,3	2,2	1,1
bn2	32x32x32			
conv3	64x32x32	3,3	1,1	1,1
bn3	64x32x32			
conv4	64x16x16	3,3	2,2	1,1
bn4	32x16x16			
conv5	128x16x16	3,3	1,1	1,1
bn5	128x16x16			
conv6	128x8x8	3,3	2,2	1,1
bn6	128x8x8			
conv7	256x8x8	3,3	1,1	1,1
bn7	256x8x8			
conv8	256x4x4	3,3	2,2	1,1
bn8	256x4x4			
conv9	512x4x4	3,3	1,1	1,1
bn9	512x4x4			
conv10	1	4,4	1,1	0,0

(b) Network structure of RecDisc4.

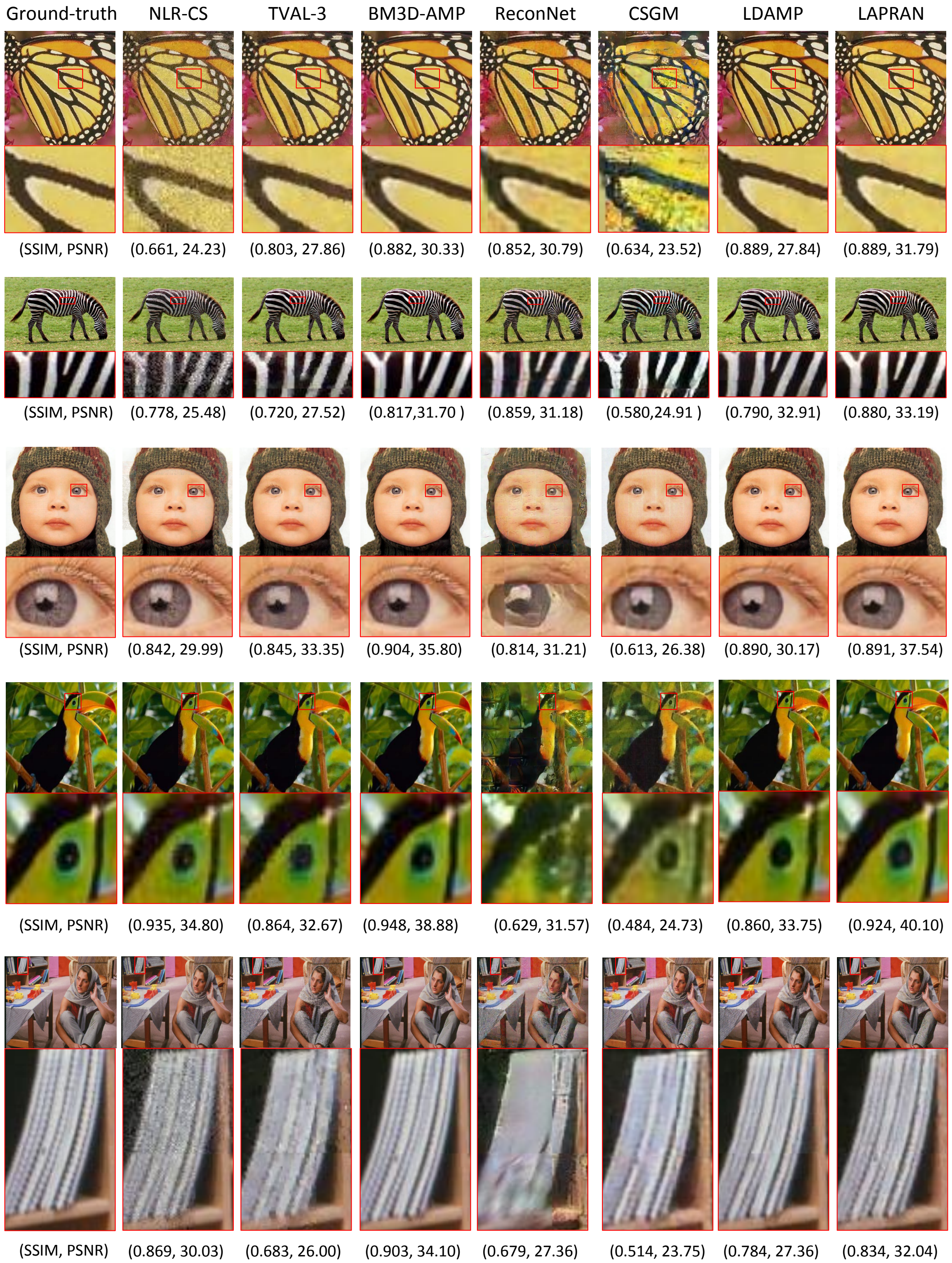


Figure A.1: Visual comparison on Set 5 and Set 14 at the CR of 5.

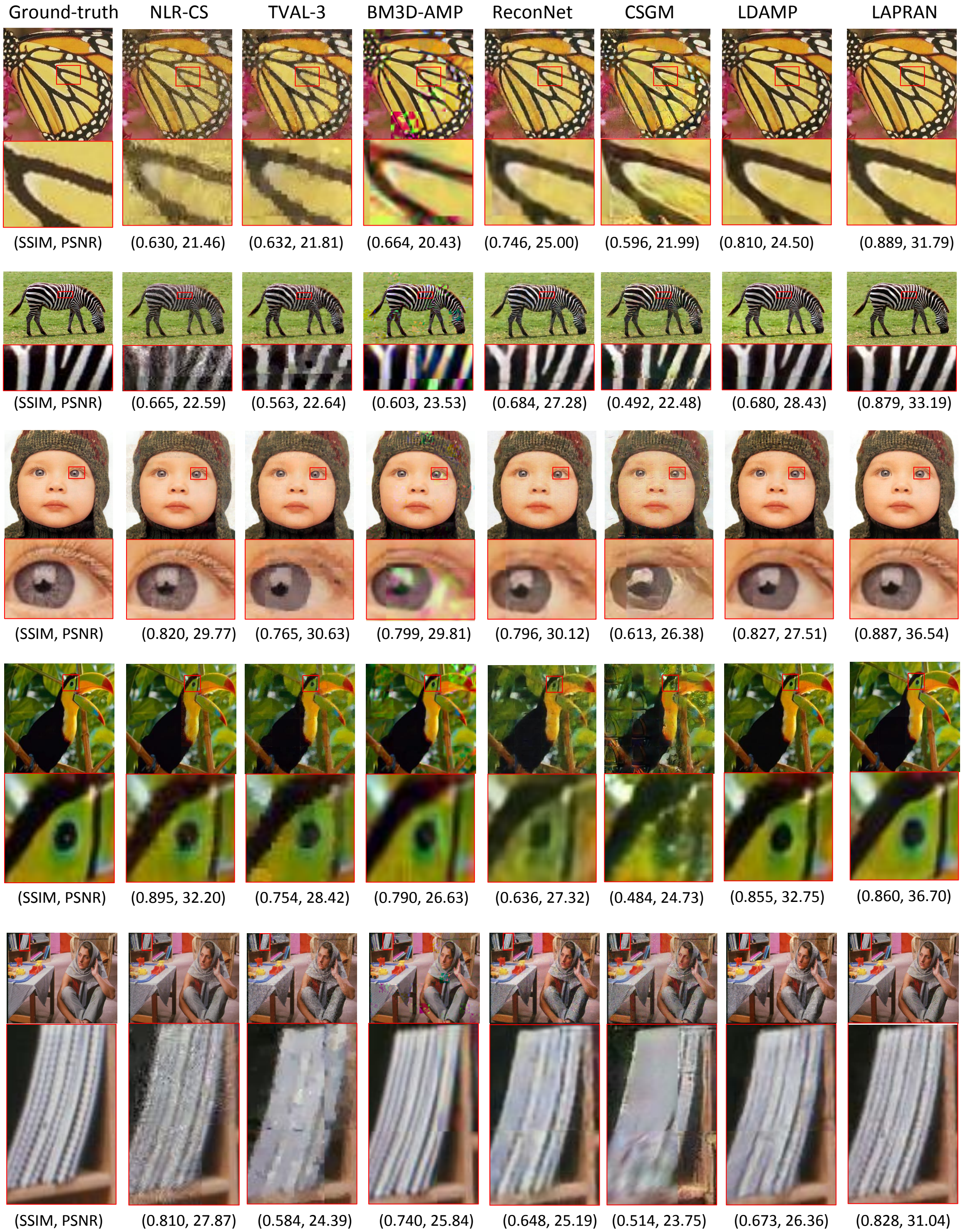


Figure A.2: Visual comparison on Set 5 and Set 14 at the CR of 10.

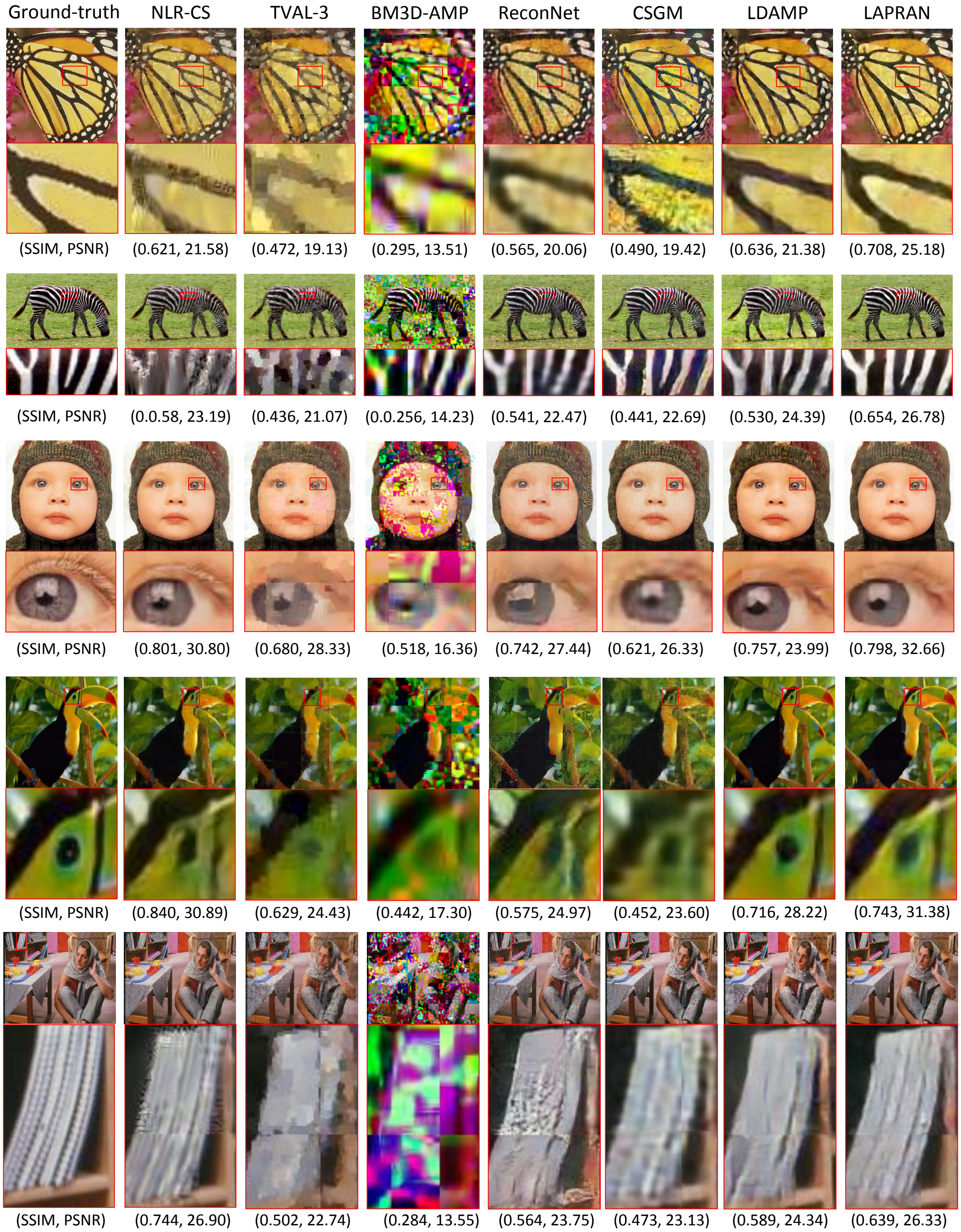


Figure A.3: Visual comparison on Set 5 and Set 14 at the CR of 20.

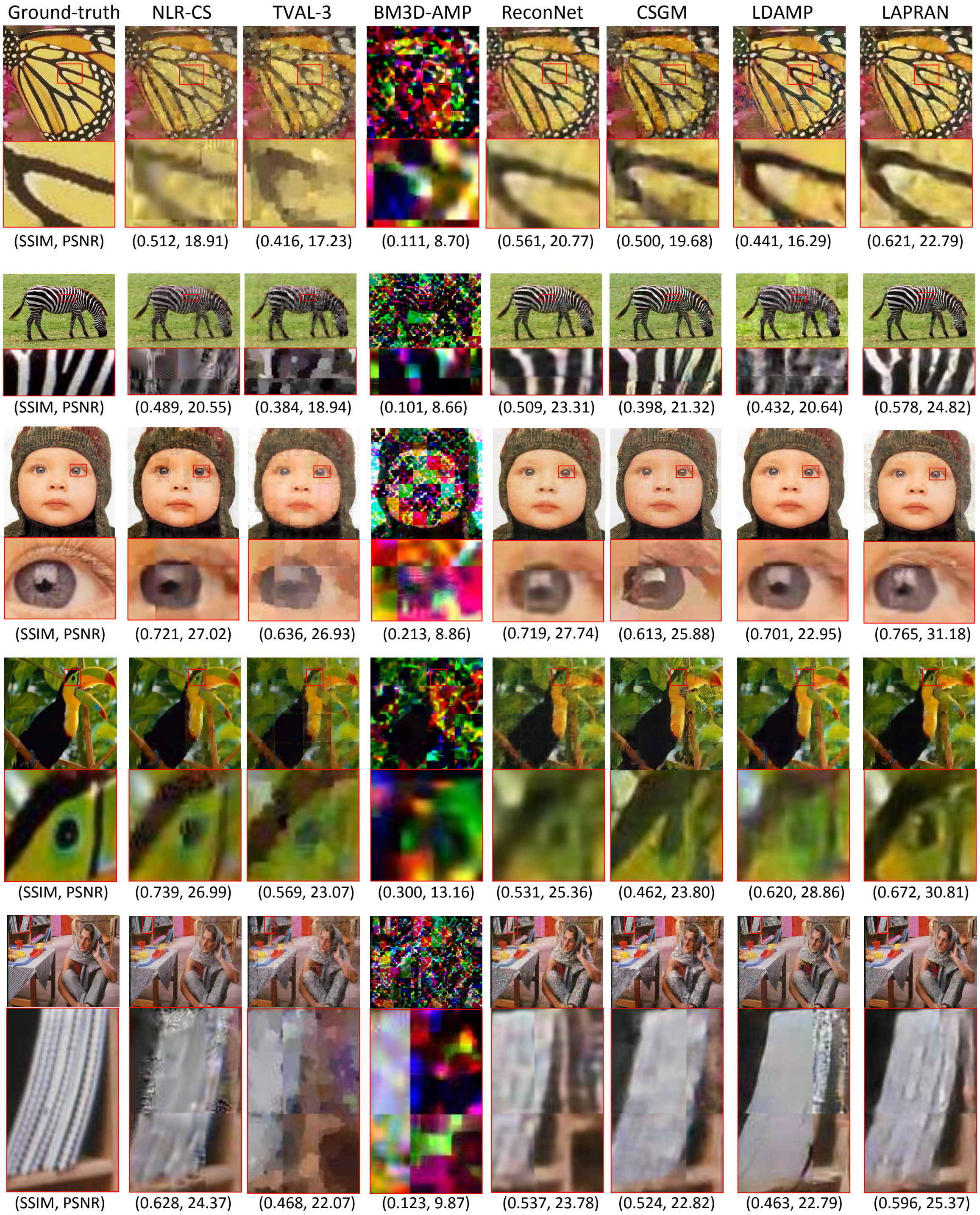


Figure A.4: Visual comparison on Set 5 and Set 14 at the CR of 30.

APPENDIX B

APPENDIX TO CSVIDEONET

In section five of the main paper, we show comparisons of our proposed "CSVideoNet" and the reference algorithms in terms of reconstruction quality from noiseless and noisy inputs. Here, we present additional reconstruction results using each approach.

The reconstruction results from noiseless CS measurements at the CRs of 25, 50, and 100 are shown in Figure B.1.

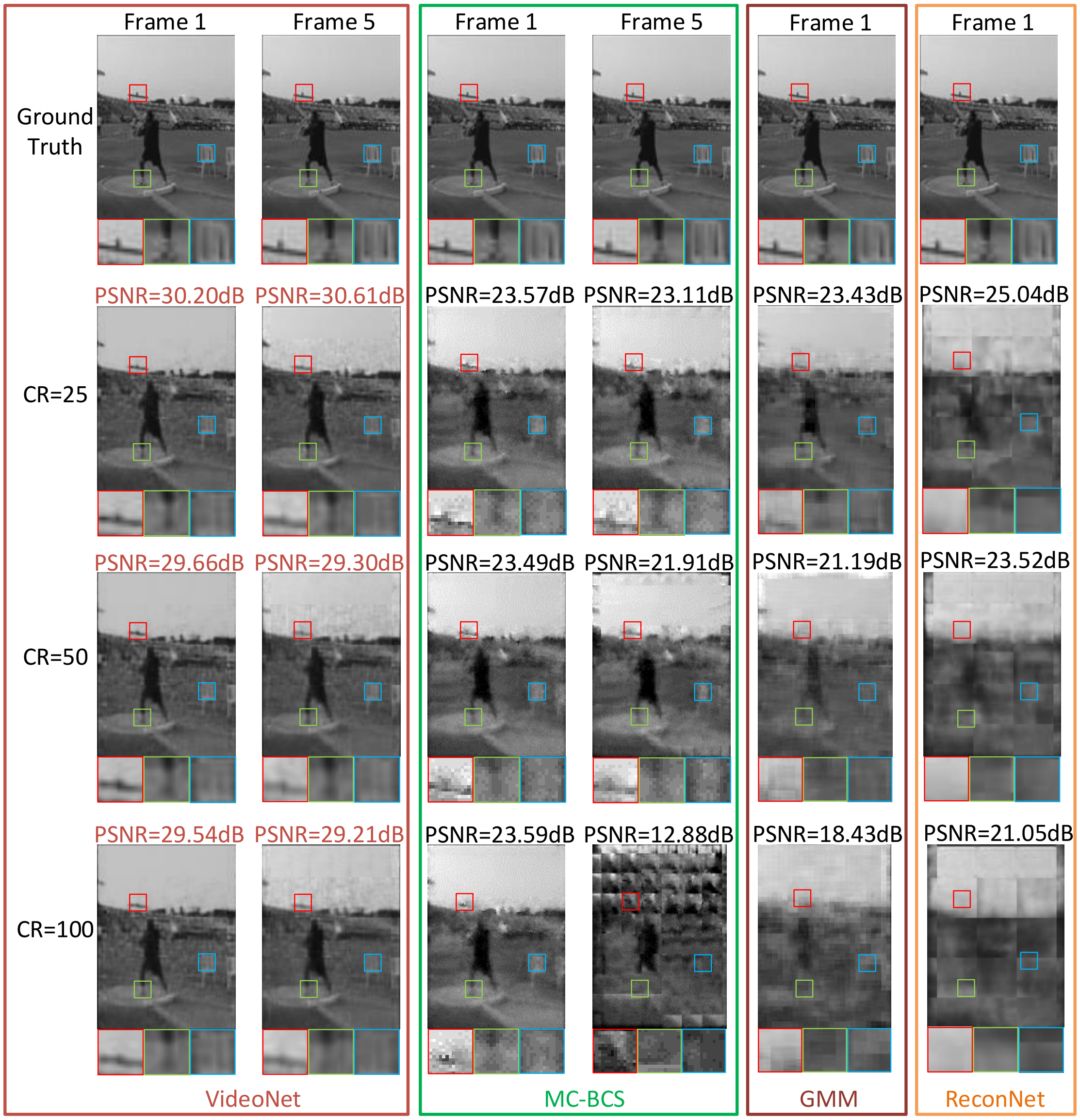


Figure B.1: Reconstruction result for each method at the CRs of 25, 50, and 100. Some visual details that are well captured by CSVideoNet but not the reference methods are highlighted for comparison purposes.

The reconstruction results from noisy CS measurements at the CRs of 25, 50, and 100 are shown in Figure B.2, B.3, and B.4, respectively.

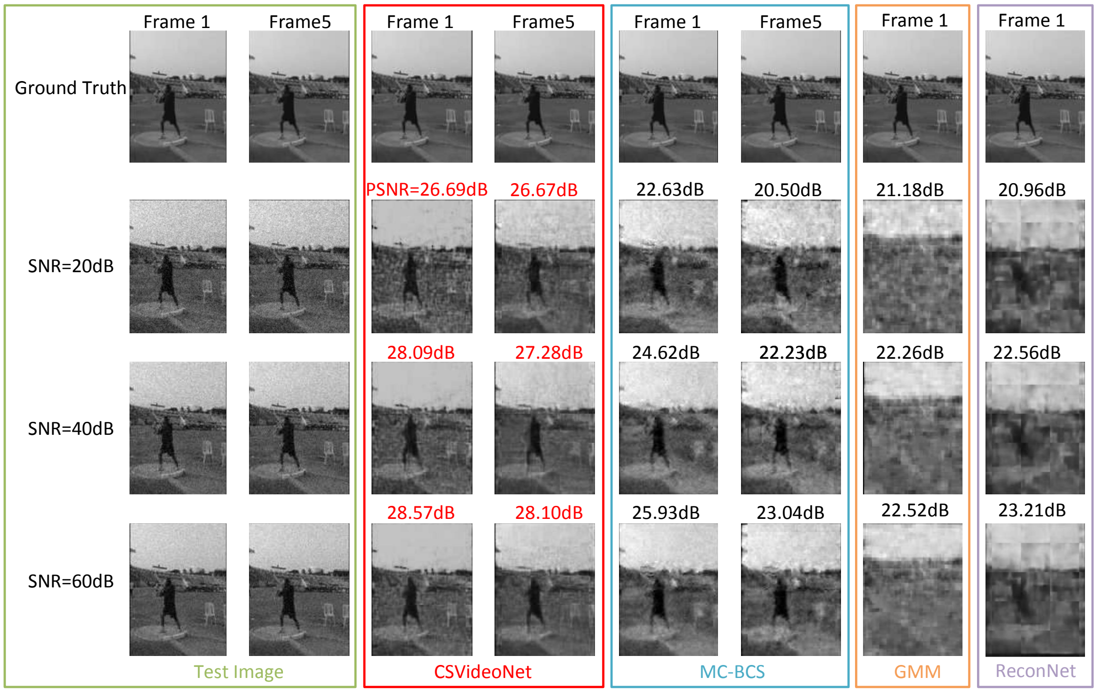


Figure B.2: Reconstruction result for each algorithm at the CR of 25 when inputs are contaminated by Gaussian white noise with the SNRs of 20dB, 40dB and 60dB.

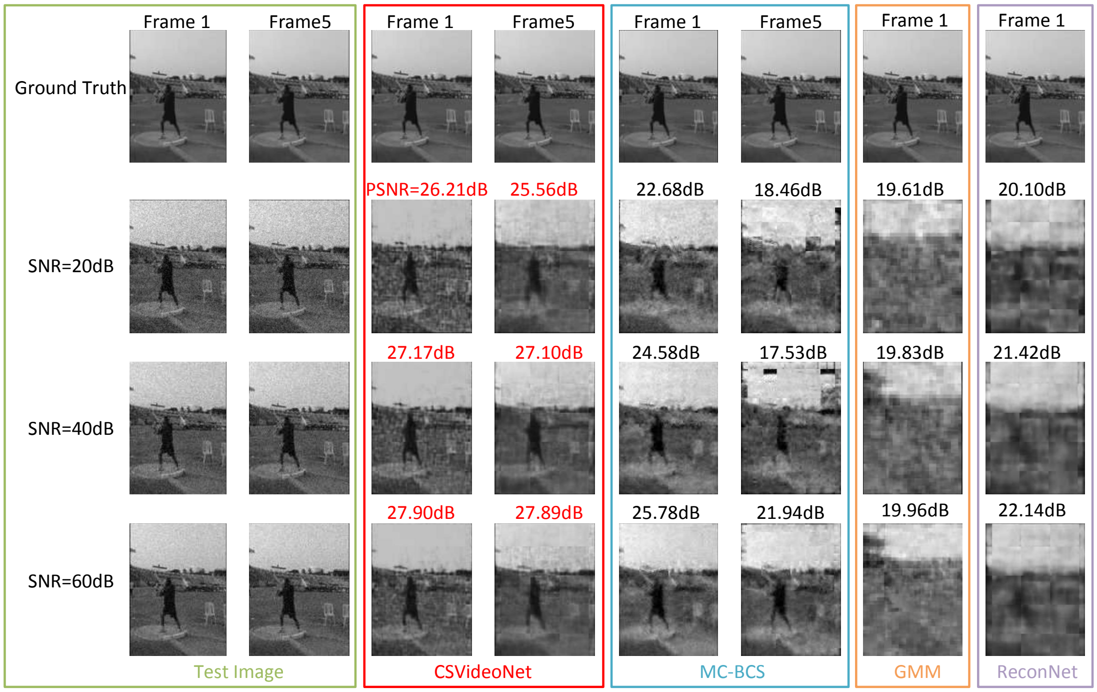


Figure B.3: Reconstruction result for each algorithm at the CR of 50 when inputs are contaminated by Gaussian white noise with the SNRs of 20dB, 40dB and 60dB.

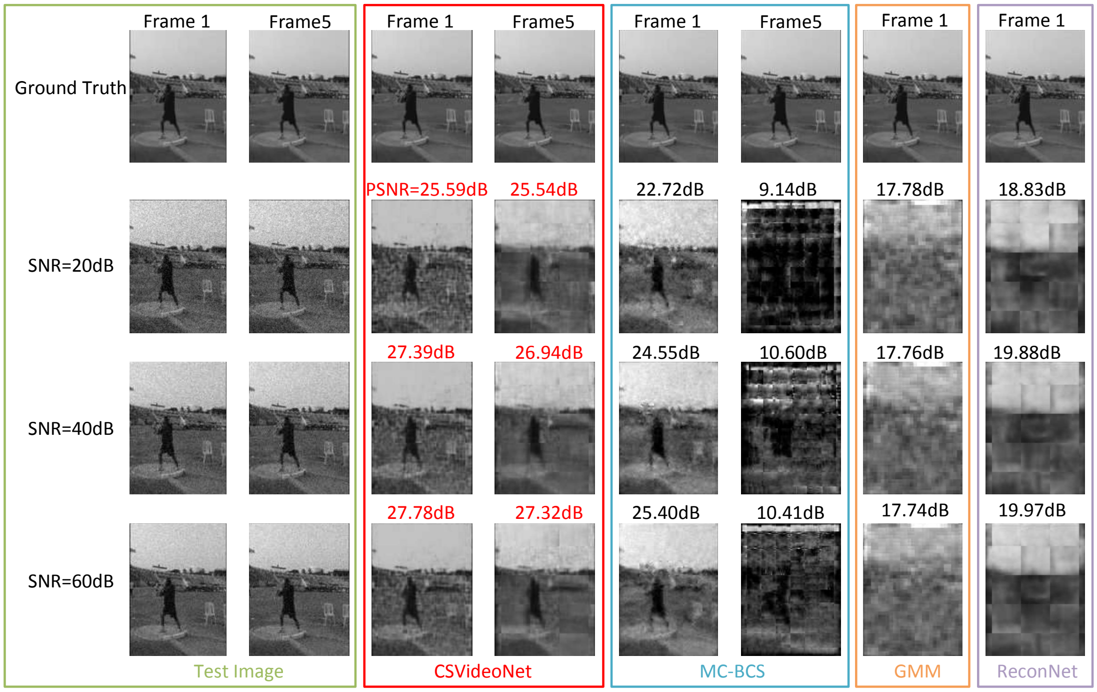


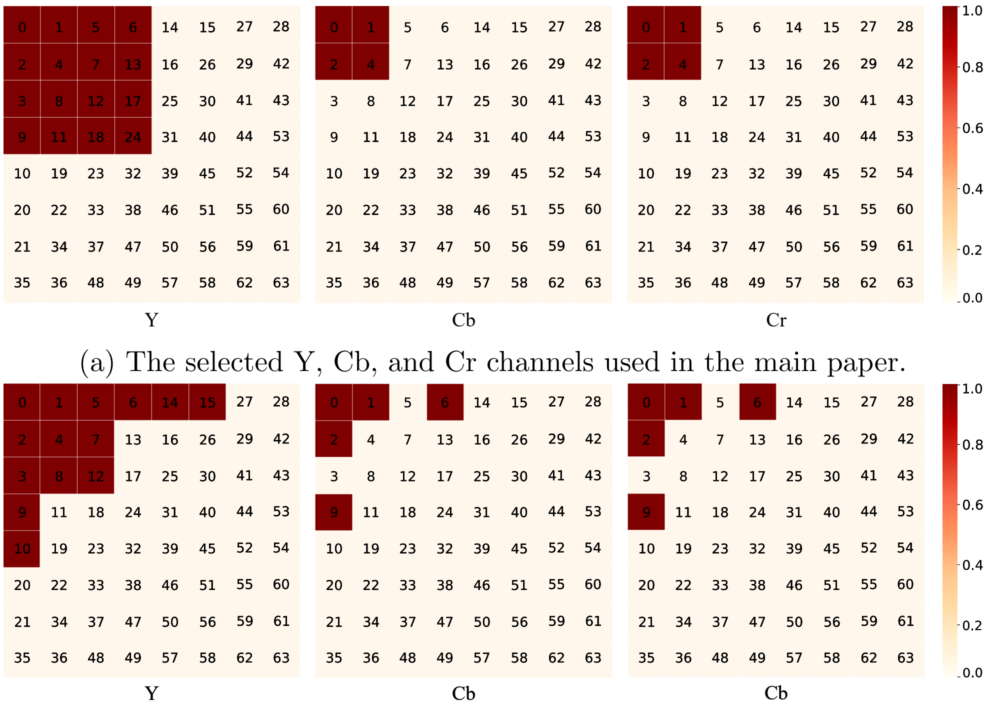
Figure B.4: Reconstruction result for each algorithm at the CR of 100 when inputs are contaminated by Gaussian white noise with the SNRs of 20dB, 40dB and 60dB.

APPENDIX C

APPENDIX TO LEARNING IN THE FREQUENCY DOMAIN

Static Channel Section

Figure C.1 shows two different channel-selection strategies for the proposed 24-channel DCT model. The selected channels in Figure C.1a is the same as we used in the main paper. The channels in Figure C.1b is selected according to the learning-based channel selection results (Figure 5 in the main paper). The classification performance comparison of the two channel-selection strategies is shown in Table C.1. The two channel-selection methods achieve similar classification accuracy. We can directly apply the channel-selection approach in Figure C.1a instead of designing more complicated strategies.



(a) The selected Y, Cb, and Cr channels used in the main paper.

(b) The selected Y, Cb, and Cr channels based on the learning-based channel selection results.

Figure C.1: A visualization of input frequency channels for the DCT-24 model on the ImageNet validation dataset. The numbers in each square represent the corresponding channel indices. The dark color indicates the current channel is selected.

Table C.1: Performance comparison of the DCT-24 models trained using the strategies in Figure C.1a and Figure C.1b. The input size of each method is normalized over ResNet-50 (RGB).

ResNet-50	#Channels	Size Per Channel	Top-1	Top-5	Normalized Input Size
DCT-24 (a)	24	56×56	76.714	93.234	0.5
DCT-24 (b)	24	56×56	76.792	93.254	0.5

Additional Instance Segmentation Results

More experiment results for instance segmentation is shown in Figure C.2.

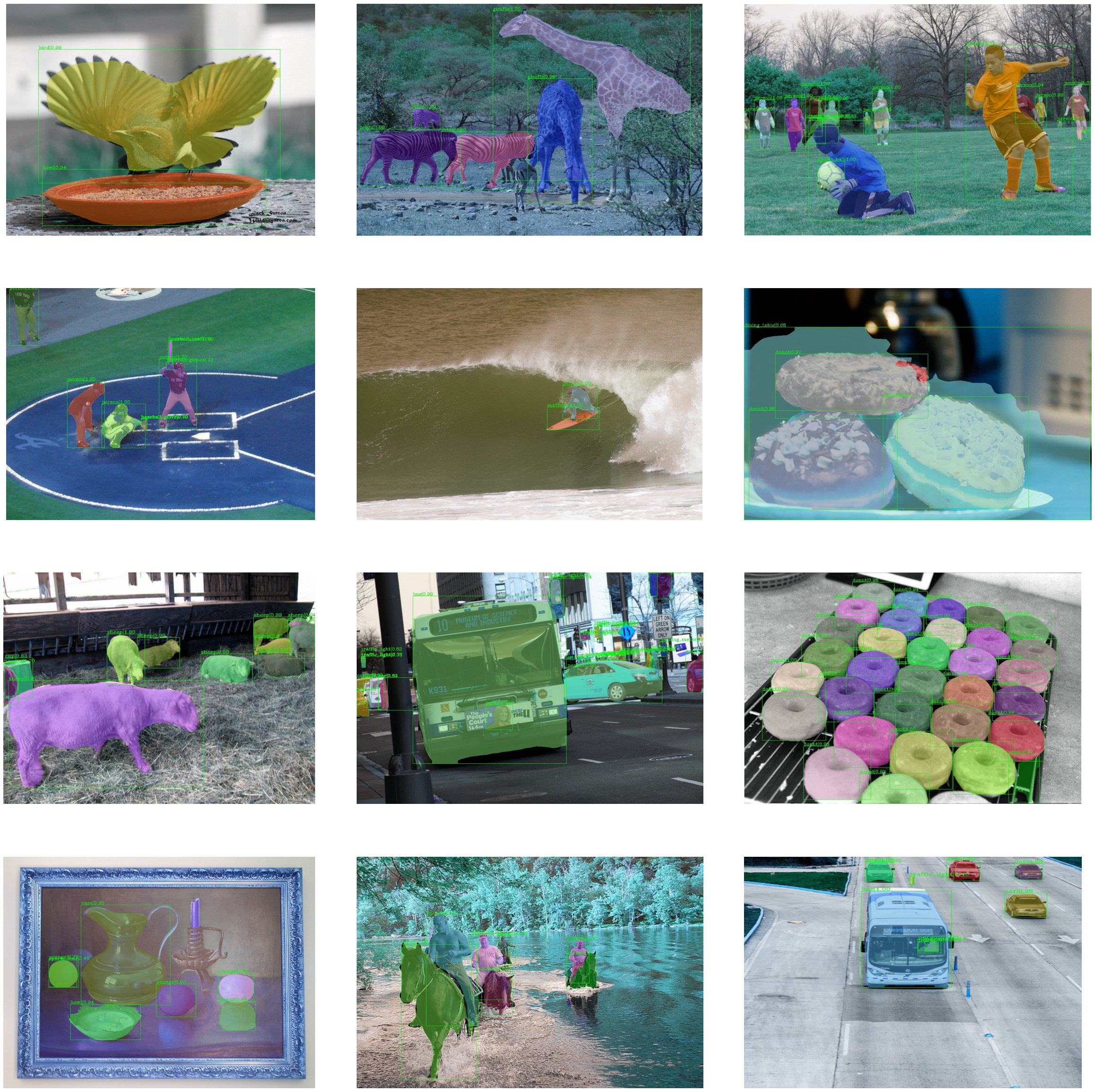


Figure C.2: Examples of instance segmentation results on the COCO dataset.

Object Detection Results

We train our model for object detection on the COCO train2017 split and evaluate on the val2017 split. Based on the Faster R-CNN Ren *et al.* (2017), our model consists of a frequency-domain ResNet-50 model (introduced in Section 4.1 in the main paper) and a feature pyramid network Lin *et al.* (2017) as the backbone. The frequency-domain ResNet-50 model is fine-tuned with the classification head and bounding box regression head. Input images are resized to a maximum scale of1600×2666 without changing the aspect ratio. The corresponding DCT coefficients have a maximum size of 200×334, which are fed into the ResNet-50-FPN for feature extraction. The rest of the configurations follow those of MMDetection Chen *et al.* (2019a).

In Table C.2, we report the results on the object detection task. The proposed method can achieve a 0.8% AP improvement compared to the baseline Faster R-CNN on the COCO dataset.

Table C.2: Bbox AP results of Faster R-CNN using different backbones on COCO 2017 validation set. The baseline Mask R-CNN use a ResNet-50-FPN as the backbone. The DCT method uses the frequency-domain ResNet-50-FPN as the backbone.

Backbone	#Channels	Size Per Channel	bbox					
			AP	AP@0.5	AP@0.75	AP_S	AP_M	AP_L
ResNet-50-FPN (RGB)	3	800×1333	36.4	58.4	39.1	21.5	40.0	46.6
DCT-24 (ours)	24	200×334	37.2	58.8	39.9	21.9	40.7	48.9
DCT-48 (ours)	48	200×334	37.1	58.6	40.2	21.7	40.9	48.8
DCT-64 (ours)	64	200×334	37.2	58.5	40.6	21.9	40.9	48.3