

PROGRAMMABLE SPATIALLY VARIANT SINGLE-
PIXEL IMAGING BASED ON COMPRESSIVE SENSING

SHIN ZHEN YONG

MASTER OF ENGINEERING SCIENCE

LEE KONG CHIAN FACULTY OF ENGINEERING AND
SCIENCE
UNIVERSITI TUNKU ABDUL RAHMAN
MAY 2021

TABLE OF CONTENTS

	Page
ABSTRACT	iii
ACKNOWLEDGEMENTS	v
APPROVAL SHEET	vi
LIST OF TABLES	vii
LIST OF FIGURES	xi
LIST OF ABBREVIATIONS/NOTATION/GLOSSARY OF TERMS	xvii

CHAPTER

1	INTRODUCTION	1
	1.1 Background	1
	1.2 Problem Statement	3
	1.3 Aims and Objectives	7
2	LITERATURE REVIEW	8
	2.1 Single-Pixel Imaging	8
	2.1.1 Motivation and Development of Single-Pixel Imaging	8
	2.1.2 Principles and Architecture of Single-Pixel Imaging System	8
	2.2 Compressed Sensing (CS)	12
	2.2.1 Conventional Image Compression Techniques	12
	2.2.2 The Mathematics Behind CS	16
	2.2.3 Block-Based Approach Of CS	18
	2.3 Sensing Matrices In CS	19
	2.3.1 The Properties of Sensing Matrices In CS	19
	2.3.2 Proof of The Restricted Isometry Property	22
	2.3.3 Random and Deterministic Sensing Matrices	25
	2.4 Image Recovery Algorithms Of CS	26
	2.4.1 Conventional Image Recovery Algorithms	26
	2.4.2 Applications of Convolutional Neural Network (CNN)	28
3	METHODOLOGY	30
	3.1 Introduction	30
	3.2 Single-Pixel Data Acquisition Based On CS	31
	3.2.1 CS Measurement Acquisition Process	31
	3.2.2 Choices of Sensing Matrices	32
	3.2.3 CS Measurements Acquisition Process for Bipolar Sensing Matrices	35
	3.3 Design of Programmable SVR Sensing Matrices	36
	3.4 Block-Based Framework with SVR Matrices	39
	3.5 Application of ReconNet as the CS Image Recovery Algorithm	40
	3.6 Performance Evaluations	41
4	RESULTS AND DISCUSSION	43
	4.1 Programmable Spatially Variant Resolution (SVR) Single-Pixel Imaging	43

4.1.1	Evaluation of Values Of z_0 For CPA Sensing Matrices	43
4.1.2	Comparison of Various SVR Sensing Matrices	54
4.1.3	Comparison of UR and SVR CPA Sensing Matrices	63
4.2	Block-Based SVR Single-Pixel Imaging With CS	73
4.2.1	Analysis on The Effects of Image Block Size	73
4.2.2	Comparison of SVR RBP, HMP, And CPA Sensing Matrices For Block-Based CS	83
4.2.3	Comparison of UR and SVR CPA Sensing Matrices For Block-Based CS	92
4.3	Block-Based Single-Pixel Imaging with ReconNet	101
4.3.1	The Network Training Results of ReconNet	101
4.3.2	Comparison of ReconNet And Other CS Image Recovery Algorithms	102
4.4	Image Encryption Scheme with CS Via CPA Sensing Matrices	113
5	CONCLUSIONS AND FUTURE WORKS	115
5.1	Conclusion	115
5.2	Recommendations for Future Works	121
	REFERENCES	122
	LIST OF PUBLICATIONS	129

ABSTRACT

Programmable Spatially Variant Single-Pixel Imaging Based on Compressive Sensing

Shin Zhen Yong

Single-pixel imaging techniques made imaging easier in conditions that are unfavourable to the conventional digital cameras such as the invisible wavelengths. According to Nyquist-Shannon theorem, it requires that the number of measurements must be no less than the number of image pixels for an error-free image recovery. However, acquiring more measurements in practice increases the cost and operating time which hinder the practicality of single-pixel imaging. Single-pixel imaging system based on compressive sensing (CS) makes it possible to simultaneously compress and acquire image data, thus recovers images from measurements less than the requirement stated by Nyquist-Shannon theorem. In general, the image quality is proportional to the number of measurements which contributes to the acquisition and computational time accordingly. Hence, the essential goal of efficient single-pixel imaging is to maintain a high recovered image quality while reducing the number of measurements and processing time.

In the conventional uniform resolution (UR) single-pixel imaging, all image pixels have the same size and are equally weighted. Therefore, a high pixel-to-pixel fidelity image recovery requires many measurements. However, high pixel-to-pixel fidelity recovery is not always needed in most visual tasks. This thesis proposes a programmable spatially variant resolution (SVR) technique in single-pixel imaging based on CS. In the proposed method, image

pixels are differently sized and formed higher and lower resolution regions. Since lower resolution regions require less measurements for recovery, most of the measurements are spent in the recovery of the higher resolution regions. Thus, SVR single-pixel imaging is able to maintain higher image quality with significantly fewer measurements. Since recovering large images requires longer time and more computational resources, block-based CS technique is proposed to reduce the computational cost by dividing them into small image blocks. Each image block is recovered in the same way as recovering individual images. In this project, a design of SVR sensing patterns is proposed and applied to the block-based CS which can reduce the complexity and time needed for the result computation.

Recently, many convolutional neural networks (CNN) were proposed as the CS image recovery algorithms. Conventionally, the network inputs of an CNN are images and the network outputs are the predictions of the labels of the images. In the framework where CNNs are applied as the CS image recovery algorithms, the network inputs would be the CS measurements and the network outputs would be the recovered images. The iterative nature of the conventional CS image recovery algorithms increases the computational resources needed and the time for image recovery. In comparison, CNNs are non-iterative. Many studies had shown that CNNs improve the recovered image quality and reduce the time cost for image recovery drastically. In this project, a CNN called ReconNet is adapted as the CS image recovery algorithm.

The results of this project show that the proposed SVR sensing patterns are able to improve the image quality and time efficiency for small number of measurements as compared to the conventional UR sensing patterns. The

improvements can be seen in both the conventional CS approach as well as the block-based CS approach. In addition, SVR sensing patterns are able to retain the image quality better as the number of measurements gets smaller. Furthermore, the results have also shown that compared to the other conventional CS image recovery algorithms, ReconNet significantly reduces the time needed for image recovery while maintaining a high image quality for small numbers of measurements.

Hence, the proposed SVR approach is more suitable in situations where high-fidelity pixel-to-pixel recovery is not the priority and most importantly far fewer measurements are required for a comparable image quality. Moreover, ReconNet outperforms the conventional CS image recovery algorithms. This shows that the proposed SVR approach with ReconNet is more suitable than the conventional approaches for practical cases.

ACKNOWLEDGEMENT

The author gratefully acknowledges the support of funding from UTAR Research Fund (UTARRF) under the Project No. IPSR/RMC/UTARRF/2018-C2/C05, Vote No. 6200/CE9.

**LEE KONG CHIAN FACULTY OF ENGINEERING AND SCIENCE
UNIVERSITI TUNKU ABDUL RAHMAN**

Date: 18 May 2021

SUBMISSION OF DISSERTATION

It is hereby certified that Shin Zhen Yong (ID No: 1903428) has completed this final year project/ dissertation/ thesis* entitled "Programmable Spatially Variant Single-Pixel Imaging Based on Compressive Sensing" under the supervision of Dr. Chua Sing Yee (Supervisor) from the Department of Electronic and Electrical Engineering, Faculty of Science and Engineering, and Ts. Dr. Pua Chang Hong (Co-Supervisor)* from the Department of Electronic and Electrical Engineering, Faculty of Science and Engineering.

I understand that University will upload softcopy of my dissertation in pdf format into UTAR Institutional Repository, which may be made accessible to UTAR community and public.

Yours truly,



(*Shin Zhen Yong*)

*Delete whichever not applicable

DECLARATION

I hereby declare that the dissertation is based on my original work except for quotations and citations which have been duly acknowledged. I also declare that it has not been previously or concurrently submitted for any other degree at UTAR or other institutions.

Name Shin Zhen Yong

Date 21 May 2021

APPROVAL SHEET

This dissertation/thesis entitled “**Programmable Spatially Variant Single-Pixel Imaging Based on Compressive Sensing**” was prepared by SHIN ZHEN YONG and submitted as partial fulfilment of the requirements for the degree of Master of Engineering Science / ~~Doctor of Philosophy~~ at Universiti Tunku Abdul Rahman.

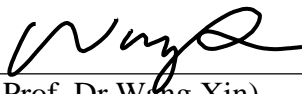
Approved by:



(Dr. Chua Sing Yee)
Date: 21 May 2021
Assistant Professor/Supervisor
Department of Electronic and Electrical Engineering
Lee Kong Chian Faculty of Engineering & Science (LKC FES)
Universiti Tunku Abdul Rahman



(Ts. Dr. Pua Chang Hong)
Date: 21/05/2021
Associate Professor/Co-supervisor
Department of Electronic and Electrical Engineering
Lee Kong Chian Faculty of Engineering & Science (LKC FES) Universiti
Tunku Abdul Rahman



(Assoc. Prof. Dr Wang Xin)
Date: 21/05/2021
Associate Professor/External Co-supervisor School of Engineering
Monash University Malaysia

LIST OF TABLES

Table		Page
2.1	Characteristics of the images acquisition and reconstruction approaches in single-pixel imaging	12
4.1	The values of PSNR (in dB) of each individual image recovered via SVR CPA sensing matrices generated with $z_0 \in \{0.19, 0.39, 0.59, 0.79, 0.99\}$ from $SR \in \{0.25, 0.1, 0.04, 0.01\}$. The highest value in each setting is in bold.	44
4.2	The values of RMSE of each individual image recovered via SVR CPA sensing matrices generated with $z_0 \in \{0.19, 0.39, 0.59, 0.79, 0.99\}$ from $SR \in \{0.25, 0.1, 0.04, 0.01\}$. The lowest value in each setting is in bold.	46
4.3	The values of SSIM of each individual image recovered via SVR CPA sensing matrices generated with $z_0 \in \{0.19, 0.39, 0.59, 0.79, 0.99\}$ from $SR \in \{0.25, 0.1, 0.04, 0.01\}$. The highest value in each setting is in bold.	48
4.4	Time cost for image recovery (in seconds) of each individual image recovered via SVR CPA sensing matrices generated with $z_0 \in \{0.19, 0.39, 0.59, 0.79, 0.99\}$ from $SR \in \{0.25, 0.1, 0.04, 0.01\}$. The lowest time cost in each setting is in bold.	50
4.5	The values of PSNR (in dB) of each individual image recovered via SVR RBP, HMP, and CPA sensing matrices. The highest value in each setting is in bold.	55
4.6	The values of RMSE of each individual image recovered via SVR RBP, HMP, and CPA sensing matrices from $SR \in \{0.5, 0.25, 0.1, 0.04\}$. The lowest value in each setting is in bold.	56
4.7	The values of SSIM of each individual image recovered via SVR RBP, HMP, and CPA sensing matrices from $SR \in \{0.5, 0.25, 0.1, 0.04\}$. The highest value in each setting is in bold.	57

4.8	The time cost for image recovery (in seconds) of each individual image recovered via SVR RBP, HMP, and CPA sensing matrices from $SR \in \{0.5, 0.25, 0.1, 0.04\}$. The lowest time cost in each setting is in bold.	58
4.9	The values of PSNR (in dB) of each individual image recovered via UR and SVR CPA sensing matrices from $SR \in \{0.5, 0.25, 0.1, 0.04\}$. The highest value in each setting are in bold.	65
4.10	The values of RMSE of each individual image recovered via UR and SVR CPA sensing matrices from $SR \in \{0.5, 0.25, 0.1, 0.04\}$ The lowest value in each setting is in bold.	66
4.11	The values of SSIM of each individual image recovered via UR and SVR CPA sensing matrices from $SR \in \{0.5, 0.25, 0.1, 0.04\}$. The highest value in each setting are in bold.	67
4.12	The time cost for image recovery of each individual image recovered via UR and SVR CPA sensing matrices from $SR \in \{0.5, 0.25, 0.1, 0.04\}$. The lowest time cost in each setting is in bold.	68
4.13	The values of PSNR (in dB) of each individual image with image block size $B \in \{64, 32, 16\}$ recovered via SVR CPA sensing matrices from $SR \in \{0.25, 0.1, 0.04, 0.01\}$, with and without applying BM3D. The highest value in each setting is in bold.	75
4.14	The values of RMSE of each individual image with image block size $B \in \{64, 32, 16\}$ recovered via SVR CPA sensing matrices from $SR \in \{0.25, 0.1, 0.04, 0.01\}$, with and without applying BM3D. The lowest value in each setting is in bold.	76
4.15	The values of SSIM of each individual image with image block size $B \in \{64, 32, 16\}$ recovered via SVR CPA sensing matrices from $SR \in \{0.25, 0.1, 0.04, 0.01\}$, with and without applying BM3D. The highest value in each setting is in bold.	77
4.16	The time cost for image recovery (in seconds) of each individual image with image block size $B \in \{64, 32, 16\}$ recovered via SVR CPA sensing matrices from $SR \in \{0.25, 0.1, 0.04, 0.01\}$, without	78

applying BM3D. The lowest time cost in each setting is in bold.

- | | | |
|------|---|----|
| 4.17 | The values of PSNR (in dB) of each individual image recovered via SVR RBP, HMP, and CPA sensing matrices from $SR \in \{0.25, 0.1, 0.04, 0.01\}$, with and without applying BM3D, for block-based approach. The highest value in each setting is in bold. | 84 |
| 4.18 | The values of RMSE of each individual image recovered via SVR RBP, HMP, and CPA sensing matrices from $SR \in \{0.25, 0.1, 0.04, 0.01\}$, with and without applying BM3D, for block-based approach. The lowest value in each setting is in bold. | 85 |
| 4.19 | The values of SSIM of each individual image recovered via SVR RBP, HMP, and CPA sensing matrices from $SR \in \{0.25, 0.1, 0.04, 0.01\}$, with and without applying BM3D, for block-based approach. The highest value in each setting is in bold. | 86 |
| 4.20 | The time cost for image recovery of each individual image recovered via RBP, HMP, and CPA SVR sensing matrices from $SR \in \{0.25, 0.1, 0.04, 0.01\}$, without applying BM3D, for block-based approach. The lowest time cost in each setting are in bold. | 87 |
| 4.21 | The values of PSNR (in dB) of each individual image recovered via UR and SVR CPA sensing matrices, with and without applying BM3D. The highest values in each setting is in bold. | 93 |
| 4.22 | The values of RMSE of each individual image recovered via UR and SVR CPA sensing matrices, with and without applying BM3D. The lowest values in each setting is in bold. | 94 |
| 4.23 | The values of SSIM of each individual image recovered via UR and SVR CPA sensing matrices, with and without applying BM3D. The highest values in each setting is in bold. | 95 |
| 4.24 | The time cost for image recovery (in seconds) of images recovered via UR and SVR CPA sensing matrices, with and without applying BM3D. The lowest time cost in each setting is in bold. | 96 |

4.25	The values of PSNR (in dB) of each individual image recovered via SVR CPA sensing matrices, through ReconNet, TVAL3, SL0, and l1-magic, from $SR \in \{0.25, 0.1, 0.04, 0.01\}$, with and without applying BM3D, for block-based approach. The highest values in each setting is in bold.	103
4.26	The values of RMSE of each individual image recovered via SVR CPA sensing matrices, through ReconNet, TVAL3, SL0, and l1-magic, from $SR \in \{0.25, 0.1, 0.04, 0.01\}$, with and without applying BM3D, for block-based approach. The lowest value in each setting is in bold.	105
4.27	The values of SSIM of each individual image recovered via SVR CPA sensing matrices, through ReconNet, TVAL3, SL0, and l1-magic, from $SR \in \{0.25, 0.1, 0.04, 0.01\}$, with and without applying BM3D, for block-based approach. The highest value in each setting is in bold.	106
4.28	The time cost for image recovery (in seconds) for each individual image recovered via SVR CPA sensing matrices through ReconNet, TVAL3, SL0, and l1-magic, from $SR \in \{0.25, 0.1, 0.04, 0.01\}$, without applying BM3D, for block-based approach. The lowest time cost in each setting is in bold.	108

LIST OF FIGURES

Figures		Page
1.1	(a) A CCD in a wire-bonded package used for ultraviolet imaging. (b) A CMOS image sensor.	5
2.1	The general setups of two single pixel imaging systems: (a) the front modulation model, (b) the back modulation model.	9
2.2	The Schematic of two mirrors from a Texas Instruments digital micromirror device (DMD).	10
2.3	(a) A standard image “peppers” in its spatial domain, (b) the same image in its DCT, and (c) the recovered image from the small amount of non-zero DCT coefficients after most of the DCT coefficients have been discarded. It can be seen from (b) that there is only a small number of DCT coefficients which are non-zero (the white spots, which cover only less than 10 % of the entire image), and that in (c) it can be seen that the changes in the image recovered after discarding most of the DCT coefficients are barely noticeable.	14
2.4	An illustration of Equation (2.2).	17
3.1	(a) Single-pixel imaging setup and process. (b) The general workflow of the complete CS measurements acquisition and image recovery processes.	31
3.2	The workflow of the CS measurements acquisition process in the simulations. The entries 0 and 1 are represented by white and black pixels, respectively.	32
3.3	Five CPA sensing matrices of size 20×50 generated with $z_0 =$ (a) 0.19 (b) 0.39 (c) 0.59 (d) 0.79 (e) 0.99. The white squares represent the entries +1 while the black squares represent the entries -1.	34
3.4	The workflow of creating a set of (a) UR and (b) SVR sensing patterns. The entries 0 and 1 are represented by black and white pixels, respectively.	38
3.5	Examples of an UR CPA sensing pattern and an SVR CPA sensing pattern. The SVR sensing pattern shown here was formed by mapping the UR sensing pattern onto an SVR grid via the transformation matrix τ . The UR sensing pattern	39

is made up of 1024 regular sized pixels and the SVR sensing pattern is made up of 363 cells. Both patterns are of the same size 32×32 .

3.6	The workflow of single-pixel imaging with CS via (a) UR and (b) SVR block-based approaches in this project.	40
3.7	The network structure of ReconNet.	41
4.1	The mean values of PSNR (in dB) of image recovered via SVR CPA sensing matrices generated with $z_0 \in \{0.19, 0.39, 0.59, 0.79, 0.99\}$ from $SR \in \{0.25, 0.1, 0.04, 0.01\}$.	52
4.2	The mean values of RMSE of image recovered via SVR CPA sensing matrices generated with $z_0 \in \{0.19, 0.39, 0.59, 0.79, 0.99\}$ from $SR \in \{0.25, 0.1, 0.04, 0.01\}$.	52
4.3	The mean values of SSIM of image recovered via SVR CPA sensing matrices generated with $z_0 \in \{0.19, 0.39, 0.59, 0.79, 0.99\}$ from $SR \in \{0.25, 0.1, 0.04, 0.01\}$.	53
4.4	The mean time cost for image recovery (in seconds) of image recovered via SVR CPA sensing matrices generated with $z_0 \in \{0.19, 0.39, 0.59, 0.79, 0.99\}$ from $SR \in \{0.25, 0.1, 0.04, 0.01\}$.	53
4.5	(Starting from left to right) The original image of “cat”, resized to 64×64 , and the images recovered via SVR CPA sensing matrices generated by $z_0 \in \{0.19, 0.39, 0.59, 0.79, 0.99\}$ from $SR = 0.5$.	54
4.6	The mean values of PSNR (in dB) of images recovered via SVR RBP, HMP, and CPA sensing matrices from $SR \in \{0.5, 0.25, 0.1, 0.04\}$.	60
4.7	The mean values of RMSE of images recovered via SVR RBP, HMP, and CPA sensing matrices from $SR \in \{0.5, 0.25, 0.1, 0.04\}$.	60
4.8	The mean values of SSIM of images recovered via SVR RBP, HMP, and CPA sensing matrices from $SR \in \{0.5, 0.25, 0.1, 0.04\}$.	61
4.9	The mean time cost for image recovery (in seconds) of images recovered via SVR RBP, HMP, and CPA sensing matrices from $SR \in \{0.5, 0.25, 0.1, 0.04\}$.	61

4.10	(First row) The original images of size 64×64 . (Second row) Images recovered via SVR RBP sensing matrix from $SR = 0.5$ with their respective SSIM. (Third row) Images recovered via SVR HMP sensing matrix from $SR = 0.5$ with their respective SSIM. (Fourth row) Images recovered via SVR CPA sensing matrix from $SR = 0.5$ with their respective SSIM.	62
4.11	Workflow of single-pixel imaging with CS via (a) UR and (b) SVR approaches in this project.	64
4.12	(From left to right) A demonstration of programming the fovea to prioritize different regions of “cat”. Notice that except for the fovea, the rest of the regions of the image is in low-resolution.	65
4.13	The mean values of PSNR (in dB) of images recovered via UR and SVR CPA sensing matrices from $SR \in \{0.5, 0.25, 0.1, 0.04\}$.	69
4.14	The mean values of RMSE of images recovered via UR and SVR CPA sensing matrices from $SR \in \{0.5, 0.25, 0.1, 0.04\}$.	69
4.15	The mean values of SSIM of images recovered via UR and SVR CPA sensing matrices from $SR \in \{0.5, 0.25, 0.1, 0.04\}$.	70
4.16	The mean time cost for image recovery (in seconds) of images recovered via UR and SVR CPA sensing matrices from $SR \in \{0.5, 0.25, 0.1, 0.04\}$.	70
4.17	(First row) The original images of size 64×64 . (Second row) Images recovered via UR CPA sensing matrix from $SR = 0.5$ with their SSIM. (Third row) Images recovered via SVR CPA sensing matrix from $SR = 0.5$ with their respective values of SSIM.	71
4.18	“baboon” and “Barbara” recovered via UR and SVR CPA sensing matrices from $SR \in \{0.5, 0.25, 0.1, 0.04\}$, shown with their respective values of SSIM.	72
4.19	The values of mean PSNR of the images with image block size $B \in \{64, 32, 16\}$ recovered via SVR CPA sensing matrices.	80
4.20	The values of mean RMSE of the images with image block size $B \in \{64, 32, 16\}$ recovered via SVR CPA sensing matrices.	80

4.21	The values of mean SSIM of the images with image block size $B \in \{64, 32, 16\}$ recovered via SVR CPA sensing matrices.	81
4.22	The mean time cost for image recovery of images with image block size $B \in \{64, 32, 16\}$ recovered via SVR CPA sensing matrices.	81
4.23	(First row) The ground truth of our testing images. From left to right, "baboon", "Barbara", "cat", "fruits", "Lena", and "peppers. All images are of size 512×512 . (Second row) The images formed from the image blocks of size 16×16 recovered via SVR CPA sensing matrices from $SR = 0.1$. (Third row) The images formed from the image blocks of size 32×32 recovered via SVR CPA sensing matrices from $SR = 0.1$. (Fourth row) The images formed from the image blocks of size 64×64 recovered via SVR CPA sensing matrices from $SR = 0.1$.	82
4.24	Mean values of PSNR (in dB) of image recovered via SVR RBP, HMP, and CPA sensing matrices from $SR \in \{0.25, 0.1, 0.04, 0.01\}$, without applying BM3D, for block-based approach.	89
4.25	Mean values of RMSE of image recovered via SVR RBP, HMP, and CPA sensing matrices from $SR \in \{0.25, 0.1, 0.04, 0.01\}$, without applying BM3D, for block-based approach.	89
4.26	Mean values of SSIM of image recovered via SVR RBP, HMP, and CPA sensing matrices from $SR \in \{0.25, 0.1, 0.04, 0.01\}$, without applying BM3D, for block-based approach.	90
4.27	Mean time cost for image recovery of image recovered via SVR RBP, HMP, and CPA sensing matrices from $SR \in \{0.25, 0.1, 0.04, 0.01\}$, without applying BM3D, for block-based approach.	90
4.28	(Start from left, first column) The original images of "baboon", "Barbara", and "cat". (Second to fourth columns) "baboon", "Barbara", and "cat" recovered via SVR RBP, HMP, CPA sensing matrices from $SR = 0.1$, without applying BM3D, for block-based approach. All images are shown with their respective values of SSIM.	91
4.29	The values of mean PSNR (in dB) of images recovered via UR and SVR CPA sensing matrices, without applying BM3D.	97

4.30	The values of mean RMSE of images recovered via UR and SVR CPA sensing matrices, without applying BM3D.	97
4.31	The values of mean SSIM of images recovered via UR and SVR CPA sensing matrices, without applying BM3D.	98
4.32	The mean time cost for image recovery of images recovered via UR and SVR CPA sensing matrices, without applying BM3D.	98
4.33	(From top to bottom row) The original images “baboon”, “Barbara”, and “cat” and their images recovered via UR and SVR CPA sensing matrices from $SR = 0.1$.	99
4.34	(Left) “peppers” recovered via SVR CPA sensing matrices from $SR = 0.04$, without applying BM3D. (Right) “peppers” recovered via SVR CPA sensing matrices from $SR = 0.04$, after applying BM3D. It can be seen that the images became less noisy and the value of SSIM increased after BM3D is applied.	100
4.35	The training and test loss of the network training for each value of SR .	102
4.36	The values of mean PSNR (in dB) of images recovered via SVR CPA sensing matrices through ReconNet, TVAL3, SL0, and l1-magic, from $SR \in \{0.25, 0.1, 0.04, 0.01\}$, without applying BM3D, for block-based approach.	110
4.37	The values of mean RMSE of images recovered via SVR CPA sensing matrices through ReconNet, TVAL3, SL0, and l1-magic, from $SR \in \{0.25, 0.1, 0.04, 0.01\}$, without applying BM3D, for block-based approach.	110
4.38	The values of mean SSIM of images recovered via SVR CPA sensing matrices through ReconNet, TVAL3, SL0, and l1-magic, from $SR \in \{0.25, 0.1, 0.04, 0.01\}$, without applying BM3D, for block-based approach.	111
4.39	The mean time cost for image recovery of images recovered via SVR CPA sensing matrices through ReconNet, TVAL3, SL0, and l1-magic, from $SR \in \{0.25, 0.1, 0.04, 0.01\}$, without applying BM3D, for block-based approach.	111
4.40	(From top to bottom row) The original images “baboon”, “Barbara”, and “cat” and their images recovered via SVR CPA sensing matrices through ReconNet, TVAL3, SL0, and l1-magic from $SR = 0.1$, without applying BM3D, for block-based approach. All images are shown with their respective values of SSIM.	112

4.41 “peppers” recovered via SVR CPA sensing matrices 114
generated with $z_0 =$ (a) 0.19 (b) 0.39 (c) 0.59 (d) 0.79 (e)
0.99. It can be seen that except for the right key $z_0 = 0.19$,
all other keys failed to recover the original image.

LIST OF ABBREVIATIONS/NOTATION/GLOSSARY OF TERMS

CS	Compressed sensing
RIP	Restricted isometry properties
NSP	Null space properties
SVR	Spatially-variant resolution
RBP	Randomly generated binary patterns
HMP	Hadamard patterns
CPA	Chaotic pattern array
PSNR	Peak signal-to-noise ratio
RMSE	Root mean squared error
SSIM	Structural similarity index measure

CHAPTER 1

INTRODUCTION

1.1 Background

Human beings have always been visual animals. The ability of visual perception has given human beings a great number of advantages that are crucial for their survival. Given the importance of the visual abilities, it is no wonder that human beings have been developing many imaging technologies such as microscopes or telescopes which can extend their visual reach. Even to this day, imaging technologies are still playing an important role in the daily lives and have helped in making lots of discoveries that are beyond imagination in the past.

Single-pixel imaging serves as an important alternative to the conventional cameras but there are still unresolved problems and limitations. Image quality, acquisition, and computational time remain the key concerns. Naturally, one of the objectives of single-pixel imaging with compressed sensing (CS) is to maintain a high image quality for the recovered image while reducing the number of measurements as much as possible. In the single-pixel imaging with CS scheme, images of the target scenes under view are acquired and subsequently recovered via a set of sensing patterns. Randomly generated sensing matrices have always been the typical choices in CS image acquisition and recovery processes because of their high probability to possess the restricting properties imposed by CS (Baraniuk et al., 2008; Candes, 2008; Donoho, 2006; Baraniuk, 2007). In the past decade, deterministic sensing

matrices have been heavily studied and applied because they are more convenient to be stored and implemented in practical situations than the randomly generated sensing matrices (DeVore, 2007).

Furthermore, in many of the typical visual tasks not every single detail of the target scene under view is being equally important to the viewers. As a matter of fact, most of the time the viewers are only interested in being able to recognize the object of interests in the target scene under view. However, in the conventional uniform resolution (UR) approach, all pixels of the sensing patterns have the same size and they are carrying the same weight in the image recovery process. Hence, if one wishes to improve the image resolution of the object of interests, the image resolution of the entire image has to be increased. As a result, this further increases the number of measurements for a higher resolution image recovery. The CS measurements become quite wasteful if the majority of them are only used to recover the fine details of regions of the target scene that the viewers have no interests in. For example, in many cases the viewers have no interests in the image backgrounds. Thus, the high-fidelity pixel-to-pixel recovery of the image background is actually not useful to the viewers in this case. In such situations, it would be beneficial to adaptively focus on the region of interest and acquire CS measurements for the recovery of the objects of interests only. As such, a design of sensing patterns to obtain image information by utilizing spatially variant resolution (SVR) technique is needed to optimize the balance between the image quality and the number of measurements.

CS image recovery is typically a computational demanding process. For instance, the recovery of a normal grayscale image of size 256×256 could

take up to half an hour or even freeze a typical laptop computer. Therefore, the development of block-based CS (Gan, 2007) approaches is crucial for one to reduce the computational resource expenses needed to recover one large image by dividing the image into many small image blocks. To reduce the expenses of computational resources and image recovery time, a block-based CS approach based on the proposed SVR technique is needed.

Conventionally, iterative algorithms have always been the common choices in the CS image recovery process (Meenakshi, 2015; Vujović et al., 2014; Abo-Zahhad et al., 2015). However, such iterative CS image recovery algorithms suffer from being computational resources expensive and always not the most efficient method in practical situations. Many studies have shown that CNN have been successfully adapted and applied to a great number of other computer vision tasks such as object classification and facial recognition. Recently, a plethora of CNN were adapted as the image recovery algorithms in CS for their quick image recovery and require less computational resources (Adler et al., 2016; Bo et al., 2017; Kulkarni et al., 2016; Lu and Bo, 2019; Metzler et al., 2017; Mousavi et al., 2015; Yao, H. et al., 2019; Zhang and Ghanem, 2018). Therefore, efficient reconstruction by adapting deep learning framework in the proposed SVR approach is desirable to suit the needs of practical application.

1.2 Problem Statement

In modern days, conventional consumer-grade digital cameras rely on charge-coupled device (CCD) and complementary metal-oxide-semiconductor (CMOS) technologies. The developments of CCD and CMOS technologies

have drastically matured the imaging technologies in the last few decades. For this reason, today's consumer-grade digital cameras in megapixel range have become low-cost and easier to acquire than they were in the past. Figure 1.1 shows the examples of CCD and CMOS image sensors. However, some imaging conditions such as wavelengths that are in the invisible zones of the light spectrum and situations where photons are scarce remain challenging for the consumer-grade digital cameras. These unusual conditions are inhibiting the conventional consumer-grade digital cameras from being more practical in a broader wavelength spectrum. For example, an infrared camera is costing as much as one hundred times the price of a typical consumer-grade digital camera for the same resolution range. Not to mention, the architectures of digital cameras operating in invisible zones of the light spectrum often become more complex, bulky, and burdensome. With the objectives of designing an imaging system with a simpler architecture and require lower cost, the single-pixel imaging systems were proposed (Duarte et al., 2008; Edgar et al., 2019). The development of single-pixel imaging techniques allows one to acquire and recover images in a broader wavelength spectrum and in conditions that are unfavorable to conventional consumer-grade digital cameras.

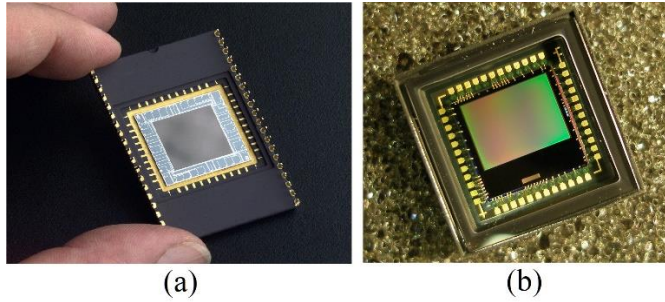


Figure 1.1: (a) A CCD in a wire-bonded package used for ultraviolet imaging (source: <https://commons.wikimedia.org/w/index.php?curid=>). (b) A CMOS image sensor (source: By Filya1 - Own work, CC BY-SA 3.0, <https://commons.wikimedia.org/w/index.php?curid=6304562>).

Traditionally, image sampling process is governed by the well-known Nyquist-Shannon theorem (Baraniuk, 2007; Candès, 2006; Cohen et al., 2009; Donoho, 2006; Duarte et al., 2008). According to Nyquist-Shannon theorem, one is required to sample a signal at a sampling rate two times the signal's highest frequency (also known as the Nyquist rate) in order to permit a successful error-free signal recovery. Nyquist-Shannon theorem governs almost all signal sampling approaches including audio and visual electronics, medical imaging devices, radio receivers, and more. In terms of two-dimensional images, one would need to acquire at least as many measurements as the number of image pixels in order to successfully recover the images without introducing errors (Duarte et al., 2008). However, in many practical cases, both the financial cost as well as the time cost for the image acquisition process increase when the number of measurements is increased. For instance, if one is to acquire more measurements in magnetic resonance imaging (MRI), the patient will have to stay in the scanners for a longer timeframe, which could be a big inconvenience and potentially making the patient feels uncomfortable. These undesirable

factors are stalling single-pixel imaging technologies from being applied more widely in practical cases. In such situations, an image sensing approach that allows one to accurately recover an image from only a small number of image data is desirable. Fortunately, based on compressed sensing (CS) mathematical theories and algorithms, one can compress and acquire images at the same time, and subsequently recover the images from less measurements than that required by the Nyquist-Shannon theorem without introducing errors (Candès, 2006; Cohen et al., 2009; Candès, 2006). The loss of image information does not affect the CS image recovery process.

In single-pixel imaging with CS scheme, images of the target scenes under view are acquired and subsequently recovered via a set of sensing patterns. The sensing matrices used in CS are restricted to some conditions imposed by CS such as the restricted isometry property (RIP) (Baraniuk, 2007; Baraniuk et al., 2008; Candès, 2006; Donoho, 2006). Conventionally, randomly generated sensing matrices such as Gaussian sensing matrices with normal distribution, binary sensing matrices with Bernoulli distribution, or Fourier sensing matrices have always been the typical choices in CS image acquisition and recovery processes because of their surprisingly high probability to possess the restricting properties imposed by CS (Candès, 2008; Candès, 2006; Donoho, 2006). However, randomly generated sensing matrices are not always the best choices in practical cases because they are more difficult to be stored and implemented in many practical cases. In recent years, sensing matrices that are not randomly generated or simply known as deterministic sensing matrices (DeVore, 2007) have been gaining interests, studied extensively and applied because they are more convenient to be stored and implemented in practical

situations than randomly generated sensing matrices. The primary objective is to design a deterministic sensing matrix that possess the restricting properties imposed by CS. Due to the deterministic nature, the required computational complexity of generating a deterministic sensing matrix is much lower than that of generating a randomly generated sensing matrix. In the past, the Hadamard sensing matrix had been studied extensively and applied as a deterministic sensing matrix in single-pixel imaging with CS (Sun et al., 2017; Zhang et al., 2017). Recently, chaotic pattern array (CPA) was proposed as one of the deterministic sensing matrices (Gan et al., 2019). The motivations of applying CPA as the sensing matrices are its properties that make it much easier to be stored and implemented than the other existing deterministic sensing matrices.

1.3 Aims and Objectives

This project aims to propose a programmable single-pixel imaging technique based on CS to improve the overall performance in terms of image quality and efficiency.

Based on the problems stated above, three objectives of this project are:

1. To propose a programmable imaging framework with SVR single-pixel imaging based on CS.
2. To improve the performance of single-pixel imaging by applying SVR approach to block-based CS.
3. To improve the performance of SVR block-based approach using ReconNet deep learning framework.

CHAPTER 2

LITERATURE REVIEW

2.1 Single-Pixel Imaging

2.1.1 Motivations and Development of Single-Pixel Imaging

As much as the conventional consumer-grade cameras have improved, their practicalities are still very limiting in situations where the wavelengths are in the invisible light spectrum and the photons are scarce. Motivated to develop cameras that can operate in invisible wavelength light spectrum and scarce photon conditions with low financial costs and simpler architecture, single-pixel imaging systems were developed. Single-pixel imaging systems make imaging easier for a light spectrum broader than that of the conventional imaging methods. Since the introduction of single-pixel imaging, it has found many applications including multispectral imaging, hyperspectral imaging, infrared imaging, terahertz imaging, 3D imaging, holography, and more (Duarte et al., 2008; Edgar et al., 2019).

2.1.2 Principles and Architecture of Single-Pixel Imaging System

In a typical single-pixel imaging system, the sensing patterns can be projected onto the target scenes (referred to as the front modulation model) or passively modulate the reflections of the target scenes via the sensing patterns (referred to as the back modulation model). Figure 2.1 shows the setups of such two models.

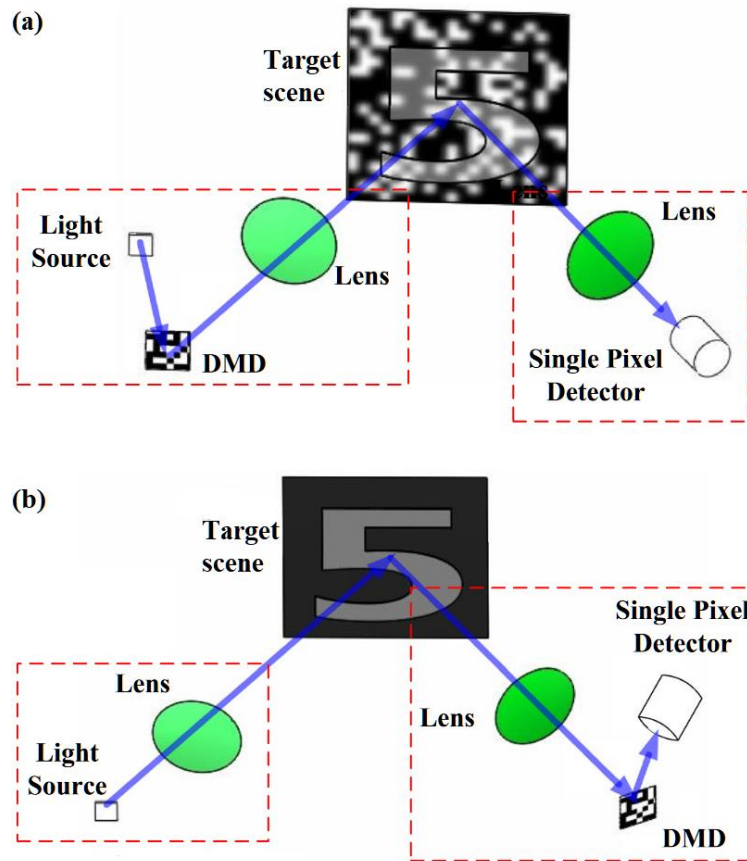


Figure 2.1: The general setups of two single pixel imaging systems: (a) the front modulation model, (b) the back modulation model (Shi et al., 2015).

In Figure 2.1, it can be seen that some of the main components of a single-pixel imaging system are lenses, a digital micro-mirror device (DMD), and a single-pixel detector. DMD is a device that has a two-dimensional array of bacterium sized and electrostatically actuated micro-mirrors. Each micro-mirror in the two-dimensional array is suspended on an individual static random access memory (SRAM) cell. The DMD can be programmed such that it orients its micro-mirrors independently to reflect the incoming light from the light source towards one of two directions ($+10^\circ$ and -10° from horizontal). When a micro-mirror is oriented so that it reflects the incoming light towards the

single-pixel detector, the position of this micro-mirror represents the entry values 1 in the sensing patterns. On the other hand, if a micro-mirror is oriented such that it reflects light away from the single-pixel detector, the position of this micro-mirror represents the entry values 0 in the sensing patterns. By switching the orientations of the micro-mirrors, different sensing patterns can be generated via the DMD. To generate values between 0 and 1, the micro-mirrors can be dithered back and forth during the image acquisition process. In order to generate bipolar sensing patterns that consist of positive and negative values such as + 1 and - 1, micro-mirrors in the positive values positions can be first all switched to “on”, keep the micro-mirrors in the negative values positions to “off”, and acquire the measurements. In the next step, all the micro-mirrors in negative values positions can be switched to "on", keep the micro-mirrors in the positive values positions to “off”, and acquire the measurements. Finally, subtract the measurements acquired in the second step from the measurements acquired in the first step. The results are the actual measurements needed. Figure 2.2 shows the schematic of two mirrors from a Texas Instruments DMD.

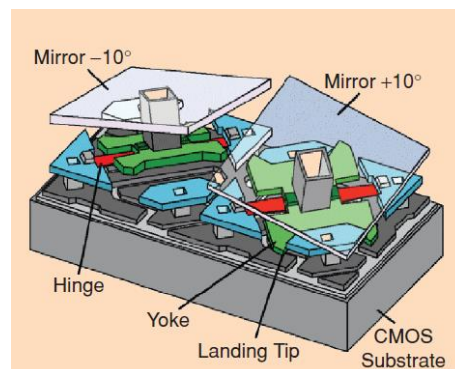


Figure 2.2: The Schematic of two mirrors from a Texas Instruments digital micromirror device (DMD) (source: http://www.optique-ingenieur.org/en/courses/OPI_ang_M09_C02/co/Contenu_31.html).

As it can be seen in Figure 2.1, a lens is used to expand the sensing patterns in order to cover the entire target scene under view. Then, another lens will be used to collect the reflected light from the target scene under view into the single-pixel detector. The single-pixel detector records the light intensity of the target scenes after they have been modulated with the sensing patterns and convert them into electrical voltage. Different from the conventional imaging system architecture, the single-pixel imaging systems have a modular design. Thus, the components of a single-pixel imaging system can be adaptively swapped out according to the objectives of the tasks. For example, some common choices of single-pixel detector include a photomultiplier tube or an avalanche photodiode for scarce photons imaging, a sandwich of multiple photodiodes with high sensitivity to different wavelengths for multimodal image sensing, a spectrometer for hyperspectral imaging, and so on (Duarte et al., 2008; Edgar et al., 2019). Further, the quantum efficiency of a single-pixel detector is higher than that of CCD and CMOS, and the fill factor of a DMD is capable of reaching up to 90% while that of CCD and CMOS is only approximately 50%. Such properties allow a single-pixel detector to receive about at least half the number of image pixels times more photons than CCD and CMOS. Hence, a single-pixel detector is capable of reducing the image distortion from dark noises and read-out noises.

Single-pixel imaging system design have successfully reduced the size, complexity and cost that would be required by conventional imaging systems. In addition, single-pixel imaging systems have also been successfully applied to areas other than two-dimensional imaging such as three-dimensional imaging and video imaging (Duarte et al., 2008; Edgar et al., 2019). The modular design

of single-pixel imaging systems also made it possible to vary the computational power by swapping out the digital computers used in the image recovery process. Hence, with more powerful digital cameras it will be possible to access higher computational power. In single-pixel imaging approach, there exist several image reconstruction methods that can recover the original images from the image data acquired. Some of the conventional image acquisition and image recovery approaches for single-pixel imaging are basis scan, adaptive basis scan, and CS with l_0 - and l_1 -minimization (Edgar et al., 2019). Each of these approaches has its own advantages and disadvantages. Table 2.1 shows the characteristics of each of the approach. This project is focusing solely on single-pixel imaging with CS approach.

Table 2.1: Characteristics of the images acquisition and reconstruction approaches in single-pixel imaging (Edgar et al., 2019).

Approach	Sensing patterns	Image reconstruction methods	Number of measurements needed
Basis scan	Basis	Inverse transform	Large
Adaptive basis scan	Basis	Inverse transform	Small
CS	Sensing matrix	l_0 - or l_1 -minimization	Small

2.2 Compressed Sensing (CS)

2.2.1 Conventional Image Compression Techniques

As imaging technologies have been continuously improved, the number of images that needs to store today has reached astronomical values. Moreover, the exponentially increasing resolution of the images also increases the size of

the digital image files. For instance, a typical image in raw format captured by a consumer-grade digital single-lens reflex (DSLR) camera has a file size that is ranging from 20 to 40 MB. Meanwhile, the typical storage space of a hard disk drive ranges from 500 GB to 4 TB. So, there exists a tremendous number of large digital image files and very limited storage space.

In order to alleviate the burden of storing such an enormous number of large digital image files, modern transform coders such as JPEG (Joint Photographic Experts Group) are applied to reduce the image file size by compressing the images. An image is compressed by only keeping the small amount of significant data of the image and discarding the majority of the image data. Counterintuitively, even though the amount of significant data of an image that is left is very minimal, the changes in the overall image quality of the image are barely noticeable after the image compression. The reason image compression is possible is that many of the normal images are sparse (which means that there are only a few nonzero coefficients) or compressible (which means that there are only a few large coefficients) when they are represented by some fixed bases (Baraniuk, 2007; Candès, 2006; Donoho, 2006). For example, many images are sparse in the well-known Fourier transform, wavelet transform, Hadamard transform, discrete cosine transform (DCT), and more (Baraniuk, 2007; Cohen et al., 2009; Donoho, 2006; Zhang et al., 2017). In fact, the famous JPEG transform coder relies on the fact that the amount of non-zero wavelet coefficients and DCT coefficients of many normal images are typically very minimal. Figure 2.3 shows a standard image “peppers” in its spatial domain, the same image in its DCT, and the recovered image from the small amount of non-zero DCT coefficients after most of the DCT coefficients have

been discarded. It can be seen from Figure 2.3(b) that after setting most DCT coefficients to zero, there is only a small number of DCT coefficients which are non-zero (the white spots, which cover only less than 10 % of the entire image), and that in Figure 2.3 (c) the changes in the image recovered after discarding most of the DCT coefficients are visually barely noticeable. Hence, the image has a sparse representation in DCT, and it can be recovered from the small amount of non-zero DCT coefficients.

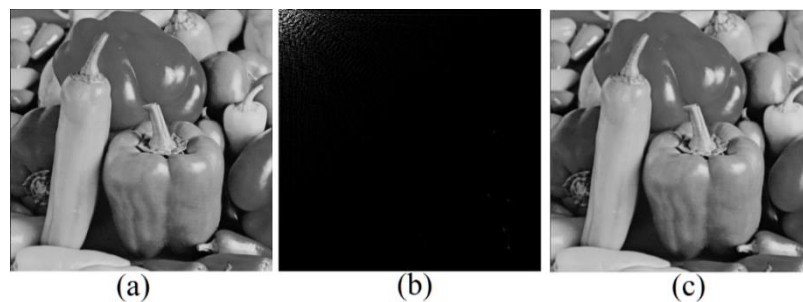


Figure 2.3: (a) A standard image “peppers” in its spatial domain, (b) the same image in its DCT, and (c) the recovered image from the small amount of non-zero DCT coefficients after most of the DCT coefficients have been discarded. It can be seen from (b) that there is only a small number of DCT coefficients which are non-zero (the white spots, which cover only less than 10 % of the entire image), and that in (c) it can be seen that the changes in the image recovered after discarding most of the DCT coefficients are barely noticeable.

In terms of mathematics, an N -pixels image of size $\sqrt{N} \times \sqrt{N}$ can be vectorized as a column vector $\mathbf{x} = (x_1 \dots x_N)^T$ of size $N \times 1$, where x_n is the n -th pixel of the image. In order to compress the image \mathbf{x} , it is required that \mathbf{x} is

sparse when it is represented in some sparsifying basis. Hence, the image \mathbf{x} can be first represented in terms of a sparsifying basis φ ,

$$\mathbf{x} = \varphi \boldsymbol{\alpha} = \sum_{n=1}^N \alpha_n \varphi_n, \quad (2.1)$$

where $\boldsymbol{\alpha}$ is the transform coefficient vector, α_n are the transform coefficients and $\varphi_n \in \mathbb{R}^N$ are the basis vectors. The image \mathbf{x} is sparse if there exist a sparsifying basis matrix φ such that cardinality of the support of the transform coefficient vector $|\text{supp}(\boldsymbol{\alpha})| \leq K < N$, where the support of the transform coefficient vector $\boldsymbol{\alpha}$, $\text{supp}(\boldsymbol{\alpha}) = \{n: \alpha_n \neq 0\}$ is a set of indices for which the transform coefficients are not zero; and $|\text{supp}(\boldsymbol{\alpha})|$ is the number of elements of $\text{supp}(\boldsymbol{\alpha})$. Next, the values and locations of only the K significant transform coefficients will be stored and the rest of the $N - K$ transform coefficients will be discarded.

However, in practical cases, the image compressing approach that was described above is not efficient for a number of reasons. To begin with, the encoder has to calculate all of the N transform coefficients $\{\alpha_n\}_{n=1}^N$ even though just a small number K of them are needed to be stored eventually. Hence most computational resources spent in the image acquisition process were spent for nothing. Moreover, the encoder also runs into the overhead of encoding the locations and values of the K significant transform coefficients (Baraniuk, 2007; Candès, 2006; Cohen et al., 2009; Donoho, 2006).

2.2.2 The Mathematics Behind CS

In the traditional approaches of signal sampling, Nyquist-Shannon's theorem demands that the sampling rate must be at the Nyquist rate or no less than twice the maximum frequency of the signal. For some types of signals such as images which are not bandlimited, the sampling rate would be determined by the temporal or spatial resolution instead. In terms of two-dimensional images, this means that no less CS measurements than the number of image pixels are needed to be acquired, failing to do so would result in introducing errors in the image recovery process.

Recall that it had been mentioned in Section 2.1.1 that the conventional image compression approaches are not efficient. Such inefficiency could be solved if there exists an image acquisition protocol that is able to directly acquire only the significant image data in the image acquisition process. The mathematics of CS have shown that such image acquisition protocol is achievable. Many studies had proven that CS had been successfully applied to single-pixel imaging (Duarte et al., 2008; Edgar et al., 2019). Hence, the hassle of acquiring all of the image data only to discard most of them at the end can be mitigated by CS. Instead, CS directly compress and acquire the CS measurements of the image \mathbf{x} at the same time in the image acquisition process (Baraniuk, 2007; Candès, 2006; Cohen et al., 2009; Donoho, 2006).

In terms of mathematics, the CS measurements of the image \mathbf{x} can be acquired with a sensing matrix $\phi \in \mathbb{R}^{M \times N}$, $M < N$,

$$\mathbf{y} = \begin{pmatrix} y_1 \\ \vdots \\ y_M \end{pmatrix} = \phi \mathbf{x} \in \mathbb{R}^M, \quad (2.2)$$

where \mathbf{y} is the measurement vector of size $1 \times M$. Each entry of the measurement vector y_m is the m -th CS measurement. If the image \mathbf{x} is represented by a sparsifying basis φ such that $\mathbf{x} = \varphi\alpha$, then Equation (2.2) becomes

$$\mathbf{y} = \phi\varphi\alpha = \Theta\alpha, \quad (2.3)$$

where $\Theta = \phi\varphi$. Figure 2.4 shows an illustration of Equation (2.3).

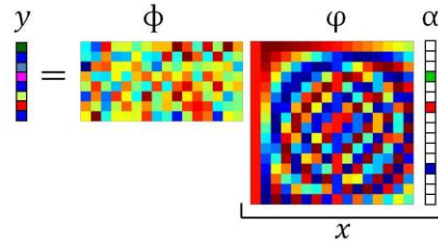


Figure 2.4: An illustration of Equation (2.2) (Baraniuk, 2007).

In practical cases, the CS measurements $\{y_m\}_{m=1}^M$ can be acquired by modulating the target scene with the projections of a set of sensing patterns via a single-pixel imaging system. Each sensing pattern is formed by reshaping each row of the sensing matrix ϕ into a square matrix of size $\sqrt{N} \times \sqrt{N}$. When the m -th sensing pattern is projected onto the target scene, the overall light intensity will be recorded by the single-pixel detector as the m -th measurement y_m .

From Figure 2.4, it can be seen that after the image acquisition via the sensing matrix ϕ , there is a loss of image information or dimensional reduction for the measurement vector \mathbf{y} is of size $1 \times M$ and the image \mathbf{x} is of size $1 \times N$, where $N > M$. Despite that, recovering \mathbf{x} from \mathbf{y} is possible via CS. Many studies had proven that CS is possible for image sampling framework provided that the images are sparse and the sensing matrix ϕ fulfills the restricting

properties imposed by CS (Baraniuk, 2007; Candès, 2006; Cohen et al., 2009; Donoho, 2006). In addition, CS possesses several properties that make it desirable. First, the CS measurements acquisition process is non-adaptive as the same sensing matrix ϕ can be used on many different images \mathbf{x} . Hence the knowledge of the images \mathbf{x} are not needed prior to the image acquisition process. Second, a small number of CS measurements can be thrown away without causing adverse effects to the image recovering process. This property makes it possible to acquire a better recovery as more measurements are acquired. Finally, only the computational complexity of CS in the image recovery process is needed to be concerned, which typically has more ample and powerful computational resources than the image acquisition process. Aside from signal or image data acquisition, CS also found other applications such as data compression, channel coding, and inverse problems (Baraniuk, 2007; Candès, 2006; Cohen et al., 2009; Donoho, 2006).

2.2.3 Block-Based Approach of CS

Despite all the advantages CS possesses, the image processing process of large images could still be very computational resources expensive. The successful recovery of large images could require a long period of image recovery time for a typical digital computer. In order to reduce the computational resources needed to recover large images, a new approach of CS called block-based CS was proposed recently (Gan, 2007). In block-based CS approach, a large image is divided into multiple smaller $B \times B$, $B < \sqrt{N}$ image blocks. Each image block is regarded as an individual image of its own and the CS measurements of each image block are acquired accordingly as

$$\mathbf{y}_i = \phi_B \mathbf{x}_i, \quad (2.4)$$

where \mathbf{y}_i is the measurement vector of the i -th image block, ϕ_B is an $M \times B^2$, $B^2 < N$ sensing matrix, and \mathbf{x}_i is the i -th image block. Hence, instead of spending more computational resources for the recovery of the entire image, multiple smaller regions of the original image which cost less computational resources can be recovered via block-based CS. Moreover, only a smaller $M \times B^2$ sensing matrix ϕ_B instead of the full $M \times N$ sensing matrix ϕ is needed to be stored. This makes both storing and implementing the sensing matrix in practice much easier. Such approach makes CS much more efficient in many practical cases.

2.3 Sensing Matrices in CS

2.3.1 The Properties of Sensing Matrices in CS

The properties of the sensing matrices ϕ are one of the most intensely studied aspects of CS (Baraniuk et al., 2008; Candes, 2008; DeVore, 2007; Gan et al., 2019; Lu et al., 2017; Sun et al., 2017). CS requires that the sensing matrix ϕ fulfills some restricting conditions. To be specific, the sensing matrix ϕ is required to possess the RIP of order K .

Definition 2.1. *For each integer $K = 1, 2, \dots$, let $\delta_K \geq 0$ be the isometry constant of ϕ such that*

$$(1 - \delta_K) \|\mathbf{x}'\|_2^2 \leq \|\phi \mathbf{x}'\|_2^2 \leq (1 + \delta_K) \|\mathbf{x}'\|_2^2 \quad (2.5)$$

holds for all K -sparse vectors $\mathbf{x}' \in \Sigma_K$. ϕ has RIP of order K provided that the isometry constant δ_K is not too close to 1 (Baraniuk et al., 2008; Candes, 2008).

The isometry constant δ_K tells one how much the length of a K -sparse vector can be changed. It can be related to the kernel of ϕ . Suppose $\delta_K < 1$, this means that the kernel of ϕ does not contain any K -sparse vectors. Assume \mathbf{x}' is K -sparse and $\phi\mathbf{x} = \mathbf{0}$, this means that $\|\phi\mathbf{x}'\|_2^2 = 0$ and $\delta_K \geq 1$, a contradiction.

Definition 2.2. Let $\phi \in \mathbb{R}^{M \times N}$ be some sensing matrix and there is an integer $K \leq \min\{M, N\}$. Then the maximum and minimum K -sparse singular values σ_{min}^K and σ_{max}^K are

$$\sigma_{min}^K = \min_{\mathbf{x}'} \frac{\|\phi\mathbf{x}'\|_2}{\|\mathbf{x}'\|_2} \quad (2.6)$$

and

$$\sigma_{max}^K = \max_{\mathbf{x}'} \frac{\|\phi\mathbf{x}'\|_2}{\|\mathbf{x}'\|_2}, \quad (2.7)$$

where \mathbf{x}' is K -sparse. The maximum and minimum K -sparse eigenvalues λ_{min}^K and λ_{max}^K are given by

$$\lambda_{min}^K = \min_{\mathbf{x}'} \frac{\|\mathbf{x}'^T \phi^T \phi \mathbf{x}'\|_2}{\|\mathbf{x}'^T \mathbf{x}'\|_2} = |\sigma_{min}^K|^2 \quad (2.8)$$

and

$$\lambda_{max}^K = \max_{\mathbf{x}'} \frac{\|\mathbf{x}'^T \phi^T \phi \mathbf{x}'\|_2}{\|\mathbf{x}'^T \mathbf{x}'\|_2} = |\sigma_{max}^K|^2 \quad (2.9)$$

The relationship between the isometry constant δ_K and the sparse eigenvalues can be shown as

$$\delta_K = \max\{1 - \lambda_{min}^K, \lambda_{max}^K - 1\}. \quad (2.10)$$

In addition to RIP, a weaker condition called the null space property (NSP) demands that $\mathcal{N}(\phi) \cap \Sigma_{2K} = \{\mathbf{0}\}$, which means that the zero vector $\mathbf{0}$ is the only $2K$ -sparse vector in the null space of the sensing matrix ϕ .

Definition 2.3. A given $\phi \in \mathbb{R}^{M \times N}$ possesses NSP of order K for $\gamma > 0$ provided

$$\|\mathbf{n}_\Lambda\|_1 \leq \gamma \|\mathbf{n}_{\Lambda^c}\|_1, \quad (2.11)$$

for all index sets Λ for which $|\Lambda| \leq K$, for all \mathbf{n} in the kernel of ϕ .

NSP guarantees the uniqueness of the solution \mathbf{x}' to Equation (2.2). Without NSP, it is always possible to find some other $2K$ -sparse vectors $\mathbf{h} = \mathbf{x}'' - \mathbf{x}' \in \Sigma_{2K}$ such that $\phi \mathbf{h} = \phi(\mathbf{x}'' - \mathbf{x}') = \mathbf{0}$ and $\phi \mathbf{x}'' = \phi \mathbf{x}'$. Thus, one would not be able to distinguish between any two potential solutions \mathbf{x}' and \mathbf{x}'' for some given measurement vector \mathbf{y} . One way to determine if the sensing matrix ϕ possesses NSP is to make sure that the spark of the sensing matrix ϕ is $\text{spark}(\phi) > 2K$, where $\text{spark}(\phi)$ is the minimum number of the columns or rows of ϕ that are linearly dependent (Baraniuk, 2007; Baraniuk et al., 2008; Candes, 2008; Candès, 2006; Donoho, 2006). But to determine whether the sensing matrix ϕ satisfies the RIP and NSP or not is normally a complex and difficult task. Instead, the sensing matrix ϕ can simply fulfill a simpler condition known as mutual coherence.

Definition 2.4. Given some sensing matrix $\phi \in \mathbb{R}^{M \times N}$, the mutual coherence of ϕ is defined as

$$\mu(\phi) = \max_{1 \leq i \neq j \leq N} \frac{|\phi(:, i)^T \phi(:, j)|}{\|\phi(:, i)\|_2 \|\phi(:, j)\|_2}, \quad (2.12)$$

where $\phi(:, i)$ is the i -th column of the sensing matrix ϕ .

The goal is to make sure that the sensing matrix ϕ has a minimum mutual coherence $\mu(\phi)$. Interestingly, Candès and Tao had discovered that if the rows of the sensing matrix ϕ are randomly chosen independently and identically distributed Gaussian vectors, then it has a surprisingly high

probability of satisfying the RIP condition (Candes et al., 2006; Candès et al., 2006; Candes and Tao, 2006).

2.3.2 Proof of The Restricted Isometry Property

Existing studies had attempted to give the acceptable conditions for random ϕ to possess the RIP (Baraniuk et al., 2008).

Suppose (Ω, ρ) is some probability space and r is some random variable of Ω . Then for some integers M and N , a random $\phi \in \mathbb{R}^{M \times N}$ can be generated by selecting the entries ϕ_{ij} as the independent realizations of r , giving a random $\phi(\omega)$, where $\omega \in \Omega^{MN}$.

Lemma 2.1. *Assume that the random matrix $\phi(\omega)$ satisfies*

$$\mathbf{P}[|\|\phi(\omega)\mathbf{x}\|_2^2 - \|\mathbf{x}\|_2^2| \geq \varepsilon\|\mathbf{x}\|_2^2] \leq 2e^{-Mc_0(\varepsilon)}, 0 < \varepsilon, \quad (2.13)$$

for all $\mathbf{x} \in \mathbb{R}^N$, where \mathbf{P} is the induced probability and $c_0(\varepsilon) > 0$ is some constant depending solely on ε . Let $K < N$, then for any set of indices Γ such that $|\Gamma| = K < N$ and any $0 < \delta < 1$,

$$1 - \delta \leq \frac{\|\phi(\omega)\mathbf{x}\|_2}{\|\mathbf{x}\|_2} \leq 1 + \delta, \quad (2.14)$$

for all $\mathbf{x} \in \mathbb{R}^N$ with support Γ , with probability at least

$$1 - 2\left(\frac{12}{\delta}\right)^K e^{-Mc_0\left(\frac{\delta}{2}\right)}. \quad (2.15)$$

Proof. As the random matrix ϕ is linear, it is only needed to consider the cases in which $\|\mathbf{x}\|_2 = 1$. Suppose $\Sigma_\Gamma \subset \mathbb{R}^N$ is the set of vectors with a support Γ . Fix δ and let $\Xi \subset \Sigma_\Gamma$ be some $\delta/4$ covering set of unit vectors for the unit vectors in Σ_Γ , that is, for all $\mathbf{x} \in \Sigma_\Gamma$ where $\|\mathbf{x}\|_2 = 1$,

$$\min_{\mathbf{v} \in \Xi} \|\mathbf{x} - \mathbf{v}\|_2 \leq \frac{\delta}{4}, \quad (2.16)$$

and $\|\mathbf{v}\|_2 = 1$ for all $\mathbf{v} \in \Xi$. It had been shown that such a set Ξ exists with a size at most $\left(\frac{12}{\delta}\right)^K$ (Baraniuk et al., 2008). By the application of the union bound, each point in the set Ξ obeys Equation (2.13) for $\varepsilon = \delta/2$ with probability

$$\left(\frac{12}{\delta}\right)^K 2e^{-Mc_0\frac{\delta}{2}}. \quad (2.17)$$

Rearranging the terms in Equation (2.11),

$$1 - \frac{\delta}{2} \leq \frac{\|\phi\mathbf{v}\|_2}{\|\mathbf{v}\|_2} \leq 1 + \frac{\delta}{2}, \quad (2.18)$$

with probability exceeds

$$1 - 2\left(\frac{12}{\delta}\right)^K e^{-Mc_0\frac{\delta}{2}}, \quad (2.19)$$

for all $\mathbf{v} \in \Xi$. Let α be the smallest number such that $\|\phi\mathbf{x}\|_2 \leq (1 + \alpha)\|\mathbf{x}\|_2$ for all $\mathbf{x} \in \Sigma_\Gamma$, it can be shown that $\alpha \leq \delta$. Since Ξ is a $\delta/4$ covering set for the unit vectors in Σ_Γ , for any unit vector $\mathbf{x} \in \Sigma_\Gamma$ there are some $\mathbf{v} \in \Xi$ such that $\|\mathbf{x} - \mathbf{v}\|_2 \leq \delta/4$. Hence, for all $\mathbf{x} \in \Sigma_\Gamma$ with $\|\mathbf{x}\|_2 = 1$, let \mathbf{v}_x be some vector such that $\|\mathbf{x} - \mathbf{v}_x\|_2 \leq \delta/4$. Then

$$\|\phi\mathbf{x}\|_2 \leq \|\phi\mathbf{v}_x\|_2 + \|\phi(\mathbf{x} - \mathbf{v}_x)\|_2 \leq 1 + \frac{\delta}{2} + (1 + \alpha)\frac{\delta}{4}. \quad (2.20)$$

As α is the smallest number such that $\|\phi\mathbf{x}\|_2 \leq (1 + \alpha)\|\mathbf{x}\|_2$ for all $\mathbf{x} \in \Sigma_\Gamma$, it must be true that

$$\alpha \leq \frac{\delta}{2} + (1 + \alpha)\frac{\delta}{4}, \quad (2.21)$$

which implies that

$$\alpha \leq \frac{3\delta}{4 - \delta} \leq \delta. \quad (2.22)$$

Hence, it proves that

$$\frac{\|\phi(\omega)\mathbf{x}\|_2}{\|\mathbf{x}\|_2} \leq 1 + \delta \quad (2.23)$$

for all $\mathbf{x} \in \Sigma_\Gamma$. In order to show the lower inequality, observe that

$$\|\mathbf{x}\|_2 \geq \|\phi\mathbf{v}_x\|_2 - \|\phi(\mathbf{x} - \mathbf{v}_x)\|_2 \geq 1 - \frac{\delta}{2} - (1 + \delta)\frac{\delta}{4} \geq 1 - \delta, \quad (2.24)$$

and the proof is complete.

Theorem 2.1. *Given M, N , and $0 < \delta < 1$. Let ϕ satisfies Lemma 2.1. Then there are some constants $c_1, c_2 > 0$ depending solely on δ such that the RIP is valid for $\phi(\omega)$ with δ and any $K \leq \frac{c_1 N}{\log(N/K)}$ with probability at least $1 - e^{-c_2 M}$.*

Proof. From Lemma 2.1 it has been shown that for any index set Γ of size K , $\phi(\omega)$ fails to satisfy the concentration inequality with probability at most

$$2 \left(\frac{12}{\delta}\right)^K e^{-Mc_0\left(\frac{\delta}{2}\right)}. \quad (2.25)$$

Since there are $\binom{N}{K} \leq \left(\frac{eN}{K}\right)^K$ such index sets, Equation (2.18) fails to hold with probability at most

$$2 \left(\frac{eN}{K}\right)^K \left(\frac{12}{\delta}\right)^K e^{-Mc_0\left(\frac{\delta}{2}\right)} = 2e^{-Mc_0\left(\frac{\delta}{2}\right) + K \log\left(\frac{eN}{K}\right) + K \log\left(\frac{12}{\delta}\right)}. \quad (2.26)$$

Hence, if $K \leq \frac{c_1 M}{\log \frac{N}{K}}$ for each fixed $c_1 > 0$, the exponent in Equation (2.26) is less

than $c_2 M$ provided that

$$c_2 \leq \frac{c_0 \delta}{2} - c_1 \left(1 + \frac{1 + \log\left(\frac{12}{\delta}\right)}{\log\left(\frac{N}{K}\right)}\right). \quad (2.27)$$

Therefore $c_1 > 0$ can always be chosen sufficiently small to ensure that $c_2 > 0$.

Therefore, with probability $1 - e^{-Mc_2}$, $\phi(\omega)$ will satisfy the concentration inequality of Equation (2.18) for each K -sparse \mathbf{x} .

Theorem 2.1 states that if some random sensing matrix obeys the concentration inequality of Lemma 2.1, it will possess RIP of order K with high probability.

2.3.3 Random and Deterministic Sensing Matrices

As it had been mentioned in Section 2.3.1 and subsequently shown in Section 2.3.2, randomly generated sensing matrices ϕ such as Gaussian sensing matrices with normal distribution, binary sensing matrices with Bernoulli distribution, or Fourier sensing matrices have a surprisingly higher chance of fulfilling the restricting properties imposed by CS. Thus, randomly generated sensing matrices ϕ have always been the common choices in CS for image acquisition and image recovery. However, randomly generated sensing matrices are not always hardware-friendly for they are more difficult to be stored and implemented in practical cases.

In the past few years, deterministic sensing matrices were proposed with the objectives of designing more hardware-friendly sensing matrices and making CS more applicable in practical cases. Some of the common choices of deterministic sensing matrices are randomly ordered Hadamard sensing matrices, Russian order Hadamard sensing matrices, and origami sensing matrices (DeVore, 2007; Yu and Liu, 2019; Zhang et al., 2017; Sun et al., 2017). It has been shown that the applications of deterministic sensing matrices have successfully reduced the required image acquisition and image recovery times, with minimal image quality trade-offs.

Recently, CPA has been proposed as one of the deterministic sensing matrices (Gan et al., 2019). Unlike most other deterministic sensing matrices,

CPA has a deterministic yet highly asymmetrical nature. Further, it is easier to store and implement CPA sensing matrices because all CPA sensing matrices can be simply generated from a seed value. Hence, the storing and implementation of CPA is much more convenient, which makes it a more suitable choice than the other deterministic sensing matrices for practical applications.

2.4 Image Recovery Algorithms of CS

2.4.1 Conventional Image Recovery Algorithms

According to linear algebra, because $M < N$, there are more unknowns than the number of equations in Equation (2.2). Hence, for some given measurement vector \mathbf{y} there will be an infinite number of solutions \mathbf{x}' subjected to $\mathbf{y} = \phi\mathbf{x}'$, and it is impossible for us to find a unique solution to Equation (2.2). However, recall that since CS only requires that the image \mathbf{x} is sparse in some sparsifying basis ϕ , so it is not needed to consider all of the potential solutions \mathbf{x}' to Equation (2.2). Instead, only the set of sparse solutions $\Sigma_K = \{\mathbf{x}': \|\mathbf{x}'\|_0 \leq K\}$ is needed to be considered, where $\|\mathbf{x}'\|_0$ is the l_0 -norm of \mathbf{x}' defined as

$$\|\mathbf{x}'\|_0 = |\text{supp}(\mathbf{x}')| \leq K \ll N. \quad (2.28)$$

More generally, the l_p -norm of \mathbf{x}' is defined as

$$\|\mathbf{x}'\|_p = \left(\sum_{n=1}^N |x'_n|^p \right)^{\frac{1}{p}}, 1 \leq p. \quad (2.29)$$

However, in many cases the solution \mathbf{x}' is not exactly sparse. So alternatively, it is more desirable when \mathbf{x}' is compressible. This means that \mathbf{x}' contains only a

small number of $K < N$ large transform coefficients and the remaining $N - K$ transform coefficients are very small or approximately zero.

Conventionally, Equation (2.2) can be solved by using the l_2 -minimization

$$\hat{\mathbf{x}}_{l_2} = \underset{\mathbf{x}'}{\operatorname{argmin}} \|\mathbf{x}'\|_2 \text{ subjected to } \phi \mathbf{x}' = \mathbf{y}. \quad (2.30)$$

Equation (2.30) has a closed form solution given by $\hat{\mathbf{x}}_{l_2} = (\phi \phi^T)^{-1} \phi^T \mathbf{y}$.

Unfortunately, such solution is almost never sparse. Therefore, alternatively, l_0 -minimization can be used to find a unique and sparse solution to Equation (2.2),

$$\hat{\mathbf{x}}_{l_0} = \underset{\mathbf{x}'}{\operatorname{argmin}} \|\mathbf{x}'\|_0 \text{ subjected to } \phi \mathbf{x}' = \mathbf{y}. \quad (2.31)$$

However, the problem with this approach is that solving Equation (2.31) is extremely computationally difficult due to l_0 -norm being discrete and non-convex (also known as NP-hard). Alternatively, instead of solving Equation (2.2) with l_0 -norm, the l_0 -norm in Equation (2.31) can be replaced with l_1 -norm then find the sparse \mathbf{x}' with l_1 -minimization (also known as basis pursuit) (Abo-Zahhad et al., 2015; Baraniuk, 2007; Candès, 2006; Donoho, 2006)

$$\hat{\mathbf{x}}_{l_1} = \underset{\mathbf{x}'}{\operatorname{argmin}} \|\mathbf{x}'\|_1 \text{ subjected to } \phi \mathbf{x}' = \mathbf{y}. \quad (2.32)$$

Equation (2.32) can be solved by applying convex optimization method or iterative greedy algorithms (Abo-Zahhad et al., 2015; Rani et al., 2018). Candès and Tao had discovered that if given an potential K -sparse solution $\mathbf{x}' \in \Sigma_K$, and there exists some positive constant $C > 0$ such that there is an upper bound to the number of measurements $M \leq CK \log(N/K)$, then the solutions to Equation (2.32) will be $\hat{\mathbf{x}}_{l_1} = \mathbf{x}$. This means that the recovery of the image $\hat{\mathbf{x}}_{l_1}$ is exact with overwhelming probability as long as the measurement vector \mathbf{y} is noiseless. In addition, in their studies, Donoho and Tanner discovered that the number of measurements M for Equation (2.32) can be determined for the exact

recovery of any given sparse image $\mathbf{x} \in \mathbb{R}^N \cap \Sigma_K$ (Donoho and Tanner, 2005).

There exists a plethora of conventional CS image recovery algorithms that had been proven to be able to successfully recover two-dimensional images from the acquired CS measurements (Abo-Zahhad et al., 2015; Candes and Romberg, 2005; Eftekhari et al., 2009; Vujović et al., 2014; Meenakshi, 2015; Li et al., 2009).

2.4.2 Convolutional Neural Networks (CNN)

The conventional CS image recovery algorithms in general have an iterative nature and require larger time cost for image recovery. In order to find an approach for which the image quality can be improved and the time cost for image recovery can be decreased, CNN have been proposed as the alternative CS image recovery algorithms.

A typical CNN consists of convolutional layers, pooling layers, and fully-connected layers. The convolutional layers map an image to its feature maps by applying kernels. Essentially, the kernels move across the image and perform element-wise multiplication with the patches of the image that they are covering. Then, the entries of the products of the element-wise multiplication will be summed and be an entry of the feature maps. Typically, the sizes of the kernels are 1×1 , 3×3 , 5×5 , or 7×7 . The pooling layers down-sample its inputs by outputting the maximum value in each of the patch of the inputs and discard all the other values with a filter usually of size 2×2 . The fully-connected layers take in 1D arrays and map them to another 1D arrays with smaller, greater, or the same dimension (Albawi et al., 2017).

Typically, a CNN takes images as inputs and return the recovered images as outputs. Since the goal of the image recovery process in CS is to recover the images from their compressed form, the conventional architecture of CNN is required to be modified. For CS image recovery tasks, the CNN first needs to take the CS measurements as inputs and produce initial estimates of the images before the convolution process. A simple approach is to add a fully-connected layer before the typical convolution layers. The fully-connected layer maps the lower dimension inputs to higher dimension vectors, followed by reshaping the higher dimension vectors to 2D arrays.

In the past, CNN have been successfully applied on plenty of computer visual tasks such as object classification, face recognition, and more (Sainath et al., 2013; Yamashita et al., 2018; Acharya et al., 2017; Lawrence et al., 1997). Unlike in conventional CNN approach where the images are the network inputs and the labels are the network outputs, in CNN for CS image recovery the CS measurements are the inputs and the recovered images are the outputs. There exist several proposed structures of CNN for CS image recovery purposes. Compared to the conventional CS image recovery algorithms, many results of the studies have shown that CNN are able to improve the image quality with significantly low time costs for image recovery compared to the other conventional CS image recovery algorithms (Adler et al., 2016; Bo et al., 2017; Kulkarni et al., 2016; Lu and Bo, 2019; Metzler et al., 2017; Zhang and Ghanem, 2018).

CHAPTER 3

METHODOLOGY

3.1 Introduction

Figure 3.1 (a) shows a typical single-pixel imaging setup and process while Figure 3.1 (b) shows the workflow of the complete CS measurements acquisition and image recovery process. In the CS measurements acquisition process, a set of sensing patterns is projected onto the target scene via a DMD. Then, the single-pixel detector measures the light reflected off the target scene and saves the values as CS measurements. After the measurements acquisition process, the sensing matrix and CS measurements are used to recover the original images with a CS image recovery algorithm. Typically, the CS image recovery algorithm solves an l_1 or TV-minimization problem to recover the images.

This chapter is organized as the following: Section 3.2 describes the simulations of single-pixel data acquisition based on CS. Section 3.3 discusses the design of programmable SVR sensing patterns. Section 3.4 explains the block-based framework of CS with SVR sensing matrices. Application of ReconNet as the CS image recovery algorithm is discussed in Section 3.5. Finally, Section 3.6 describes the metrics used to evaluate the performance of image recovered.

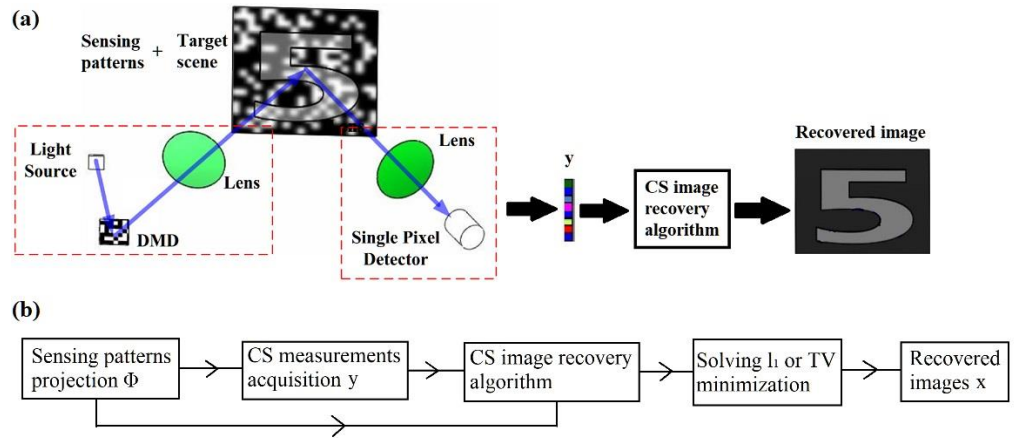


Figure 3.1: (a) Single-pixel imaging setup and process. (b) The general workflow of the complete CS measurements acquisition and image recovery processes.

3.2 Single-Pixel Data Acquisition Based on CS

3.2.1 CS Measurements Acquisition Process in the Simulation

For the numerical experiments, a total of six standard images of size 512×512 (courtesy of the Signal and Image Processing Institute at the University of Southern California) are selected in this project for testing.

In this project, the single-pixel imaging with CS process is simulated on a laptop computer with 32 GHz Intel Core-i5-7200U and 4GM RAM. In the simulations, the CS measurements are acquired according the following steps as illustrated in Figure 3.2.

- i. A sensing matrix is generated, and each row of the sensing matrix is reshaped into a set of sensing patterns.
- ii. Each of the sensing pattern are resized as the same size of the testing images.

- iii. The image pixels that are in the 0 positions of the sensing patterns are set to zero and the average values of the image pixels are calculated and saved as the CS measurements.

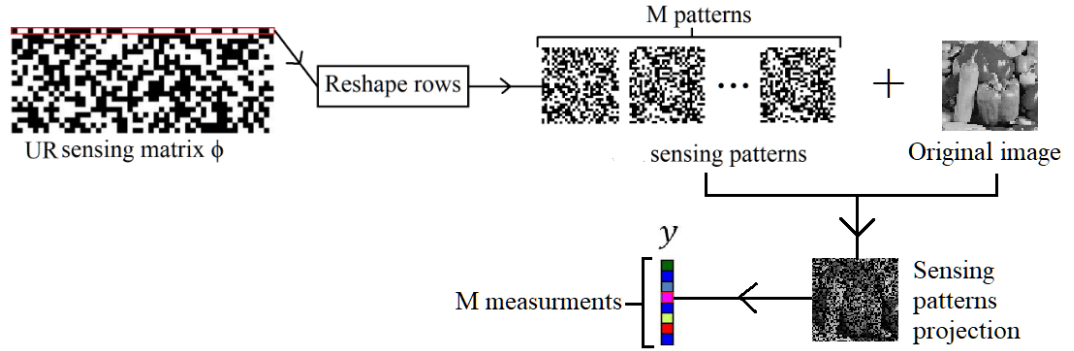


Figure 3.2: The workflow of the CS measurements acquisition process in the simulations. The entries 0 and 1 are represented by white and black pixels, respectively.

3.2.2 Choices of Sensing Matrices

In this project, three types of sensing matrices are chosen for comparison. The first sensing matrix is the randomly generated binary sensing matrix with Bernoulli distribution (RBP). RBP contains only the entries 0 and 1. The entries are abiding by the Bernoulli distribution defined as

$$P(n) = \begin{cases} 1 - p, & n = 0 \\ p, & n = 1 \end{cases}, \quad (3.1)$$

where P is the probability density function, p is the probability, and n is the event. In this project p is set as 0.5.

The second sensing matrix chosen for comparison is the deterministic Hadamard sensing matrix. The lowest order Hadamard matrix is defined as

$$H_2 = \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix},$$

and the n -th order Hadamard matrix is defined as

$$H_{2^n} = \begin{pmatrix} H_{2^{n-1}} & H_{2^{n-1}} \\ H_{2^{n-1}} & -H_{2^{n-1}} \end{pmatrix}. \quad (3.2)$$

It is worth mentioning that the originally ordered rows Hadamard sensing matrix does not satisfy the restricting properties of CS. Hence, the rows of the Hadamard sensing matrix are needed to be randomly re-ordered before reshaping them into sensing patterns.

The third sensing matrix chosen for comparison is the chaotic pattern array (CPA) sensing matrix. In terms of mathematics, CPA is based a special case of the logistic chaotic system defined as

$$z_{k+1} = 4z_k (1 - z_k), k = 0, 1, 2, \dots, \quad (3.3)$$

where $z_k \in (0, 1)$ and z_0 is a seed. With a threshold function defined as

$$T(z_k) = \begin{cases} +1, & 0.5 \leq z_k < 1 \\ -1, & 0 < z_k < 0.5 \end{cases}, \quad (3.4)$$

and its complementary function

$$\bar{T}(z_k) = 1 - T(z_k), \quad (3.5)$$

an CPA sensing matrix can be acquired as

$$\phi = \begin{pmatrix} T(z_0) & \cdots & T(z_{N-1}) \\ \vdots & \ddots & \vdots \\ T(z_{(M-1) \times N}) & \cdots & T(z_{(M \times N) - 1}) \end{pmatrix}. \quad (3.6)$$

Based on Equation (3.3 – 3.6), it can be seen that only a seed value z_0 is needed to generate an CPA sensing matrix. Therefore, unlike randomly generated and other deterministic sensing matrices for which the entire matrix needs to be saved, only the values of z_0 is needed to be saved for the implementation of different CPA sensing matrices. In addition, CPA sensing matrices also possess a desirable property of being highly asymmetrical even though they are actually

deterministic. In their works, Gan et al. had proven that CPA sensing matrices fulfill the RIP condition (Gan et al., 2019). Figure 3.3 shows five different CPA sensing matrices of size 20×50 each generated from different values of seed $z_0 \in \{0.19, 0.13, 0.59, 0.79, 0.99\}$. It can be seen that for each of the value of seed z_0 , a completely different CPA sensing matrix can be acquired, and that each CPA sensing matrix is highly asymmetrical despite being deterministic. Furthermore, the time cost required to generate a CPA sensing matrix is very short. This is true because of several properties of CPA. Firstly, the bipolar sequences (+1 and -1) that are used to generate a CPA sensing matrix is generated by a quadratic iterative equation with a hard threshold equation. Hence, complex calculations are not required in the process. Secondly, the generation of CPA sensing matrices only require the values of seed z_0 and some simple row-by-row arrangements, so its implementations are relatively simpler and faster. Such properties make CPA sensing matrices the more desirable choices compared to other randomly generated and deterministic sensing matrices in single-pixel imaging settings. In addition, the properties of CPA sensing matrices also make them suitable for image encryption and decryption with single-pixel imaging.



Figure 3.3: Five CPA sensing matrices of size 20×50 generated with $z_0 =$ (a) 0.19 (b) 0.39 (c) 0.59 (d) 0.79 (e) 0.99. The white squares represent the entries +1 while the black squares represent the entries -1.

The sensing patterns used in the CS measurements acquisition process can be obtained by reshaping each row of the sensing matrix of size $M \times N$ into a set of M square patterns of size $\sqrt{N} \times \sqrt{N}$.

Recently, a design of spatially-variant resolution (SVR) sensing patterns was proposed for single-pixel imaging with CS (Phillips et al., 2017; Shin et al., 2019). Inspired by the animal visual systems, in SVR sensing patterns approaches the target scenes under view are divided into higher and lower resolution regions. Because the lower resolution regions require a smaller number of CS measurements for recovery, most CS measurements that were acquired in the image acquisition process are going to be used in the recovery of the higher resolution regions. Therefore, the number of CS measurements needed to maintain a high image quality in SVR sensing patterns approaches are less than that required by the conventional UR sensing patterns approaches. The SVR single-pixel imaging approaches are more desirable in situations where a loyal pixel-to-pixel image recovery requires a large number of CS measurements, is computational resources expensive and not useful for the visual tasks.

3.2.3 CS Measurements Acquisition Process for Bipolar Sensing Matrices

Because HMP and CPA sensing matrices are bipolar sensing matrices (they only contain entries -1 and +1), the CS measurements acquisition process is needed to be modified. In the cases of bipolar sensing matrices, the sensing matrices are needed to be separated into two sets of complementary sensing matrices $\phi_+ = \frac{1}{2}(\mathbf{1} + \phi)$ and $\phi_- = \mathbf{1} - \phi_+$. Then, the rows of ϕ_+ and ϕ_-

are reshaped into two sets of complementary sensing patterns and alternatively projected onto the testing images. Since there are two sets of complementary sensing patterns, there will be two sets of measurement vectors \mathbf{y}_+ and \mathbf{y}_- . The actual CS measurements are acquired as $\mathbf{y} = \frac{1}{2}(\mathbf{y}_+ - \mathbf{y}_-)$. In this case the sensing matrix becomes $\hat{\phi} = \frac{1}{2}(\phi_+ - \phi_-)$. Hence, Equation (2.2) becomes

$$\frac{1}{2}(\mathbf{y}_+ - \mathbf{y}_-) = \frac{1}{2}(\phi_+ - \phi_-)\mathbf{x}. \quad (3.7)$$

3.3 Design of Programmable SVR Sensing Matrices

In conventional single-pixel imaging, the sensing patterns used in the image acquisition process have an UR grid. In an UR grid, all of the image pixels have the same size. In recent years, a different design of sensing patterns with SVR grid was proposed (Phillips et al., 2017; Shin et al., 2019; Shin et al., 2021).

The SVR sensing patterns were inspired by the animal vision systems. In an animal vision system, a vision field is divided into a higher resolution region and a lower resolution region. The fovea of an animal eye defines the higher resolution region and the peripheral vision defines the lower resolution region. Such vision systems are adaptive since the fovea is constantly being redirected to the objects of interest. SVR vision systems restrict the information needed to be processed by prioritizing the objects of interest with the fovea and limiting the information of the rest of the vision field such as the background with the peripheral vision. Similarly, in an SVR sensing pattern there is a higher resolution region consisting smaller image pixels surrounded by a lower resolution region consisting larger image pixels.

In the cases of UR sensing matrices, all image pixels of the recovered images are equally weighted. Hence, in order to achieve a high-fidelity pixel-to-pixel image recovery, the required number of CS measurements is typically large. However, in many visual tasks a high-fidelity pixel-to-pixel image recovery is not always necessary or useful. This is because most of the time, a basic recognition of the objects of interests is more important than the recognition of the backgrounds or the peripheral regions in a visual task. Since the lower resolution regions of the images require a smaller number of CS measurements for recovery, SVR sensing matrices are able to lower the number of CS measurements and spend most of the acquired CS measurements recovering the object of interests. Therefore, SVR sensing matrices are more favorable in situations where the number of CS measurements is limited, the objects of interest is prioritized, and a high-fidelity pixel-to-pixel image recovery is not useful.

In order to form an SVR sensing matrix ϕ_{SVR} , a binary transformation matrix τ of size $N \times N$ with only entries 0 and 1 can be designed and applied. In each of the j -th column of τ , the entries 1 map one entry of a row of an UR sensing matrix to several entries of a row of the SVR sensing matrix. The positions of the entries 1 in τ indicate the positions of an SVR grid which the regular sized pixels of the UR sensing pattern will be mapped on. Thus, an SVR sensing matrix can be formed via matrix multiplication

$$\phi_{SVR}^T = \tau \phi_{UR}^T, \quad (3.8)$$

where ϕ_{SVR}^T and ϕ_{UR}^T are the transpose of ϕ_{SVR} and ϕ_{UR} , respectively. Each row of ϕ_{SVR} can be reshaped as an SVR sensing pattern. To avoid potential confusions, the differently sized SVR image pixels are referred to as cells. The

total number, position, size, and shape of the cells in an SVR sensing pattern is configurable and determined by the designs of the binary transformation matrix τ . Figure 3.4 shows the workflow of creating a set of UR and SVR sensing patterns. Figure 3.5 shows examples of an UR CPA sensing pattern and an SVR CPA sensing pattern. The SVR sensing pattern shown here was formed by mapping the UR sensing pattern onto an SVR grid. The UR sensing pattern is made up of 1024 regular sized pixels and the SVR sensing pattern is made up of 363 cells. Both patterns are of the size 32×32 .

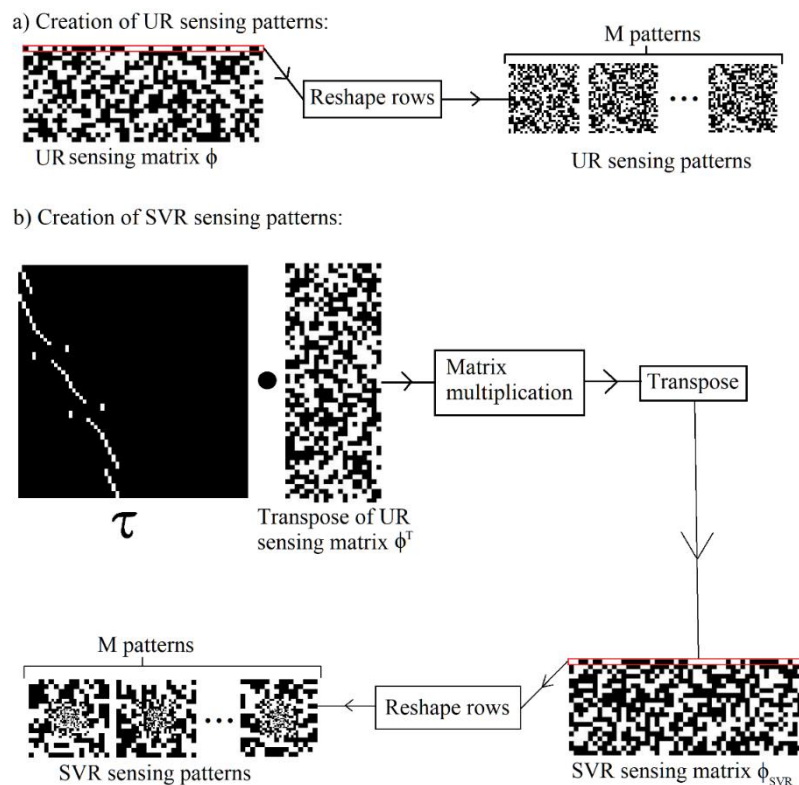


Figure 3.4: The workflow of creating a set of (a) UR and (b) SVR sensing patterns. The entries 0 and 1 are represented by black and white pixels, respectively.



Figure 3.5: Examples of an UR CPA sensing pattern and an SVR CPA sensing pattern. The SVR sensing pattern shown here was formed by mapping the UR sensing pattern onto an SVR grid via the transformation matrix τ . The UR sensing pattern is made up of 1024 regular sized pixels and the SVR sensing pattern is made up of 363 cells. Both patterns are of the same size 32×32 .

3.4 Block-Based Framework with SVR Sensing Matrices

Since the recovery of large images can be very computational expensive, block-based approach was proposed to CS. In block-based approach, the images are divided into smaller image blocks. Each image block is treated as an individual image.

Assume that the image is divided into $\mathbf{B} \times \mathbf{B}$ image blocks, in terms of block-based CS, Equation (2.2) becomes

$$y_i = \phi_B x_i, \quad (3.9)$$

where y_i is the measurement vector of the i -th image block, ϕ_B is a sensing matrix of size $M \times B^2$, and x_i is the i -th image block.

Figure 3.6 shows the workflow of single-pixel imaging with CS via (a) UR and (b) SVR block-based approaches in this project. In the simulations, an image is first divided into multiple image blocks. Then, a set of UR or SVR sensing patterns are projected onto each image block. The average of the pixel

values of each image block is saved as the CS measurements. Finally, TVAL3 is used to recover the image blocks from their CS measurements.

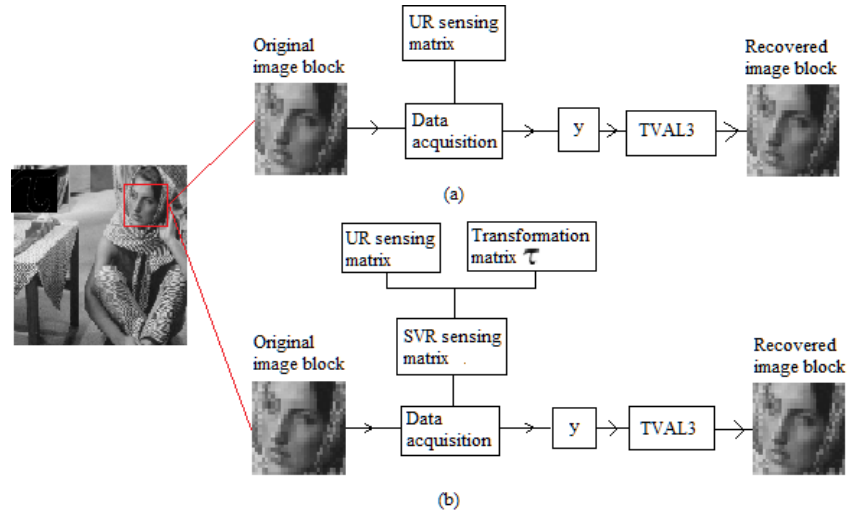


Figure 3.6: The workflow of single-pixel imaging with CS via (a) UR and (b) SVR block-based approaches in this project.

3.5 Application of ReconNet as the CS Image Recovery Algorithm

In this project, a CNN called ReconNet (Kulkarni et al., 2016) is adapted for the proposed SVR block-based CS approach. The network is trained with the CIFAR-10 image dataset. Before the network training process, a set of CS measurements of the CIFAR-10 dataset (Abouelnaga et al., 2016) is acquired via SVR CPA sensing matrices. Then, the CS measurements are used as the training and testing sets in the network training process.

Figure 3.7 shows the network structure of ReconNet. The first layer of ReconNet is a fully-connected layer, followed by several convolutional layers. All convolutional layers use ReLU after the convolution. The fully-connected layer takes block CS measurements $y_i = \phi_B x_i$ as inputs and outputs a feature

map. The first and fourth convolutional layer use 11×11 kernels and produce 64 feature maps. The second and fifth convolutional layer use 1×1 kernels and produce 32 feature maps. The third and last convolutional layers use 7×7 kernels and produce one feature map. The feature map produced by the last convolutional layer is also the recovered image block \hat{x}_i . Zero padding was used in order to maintain the feature map size constant in all layers.

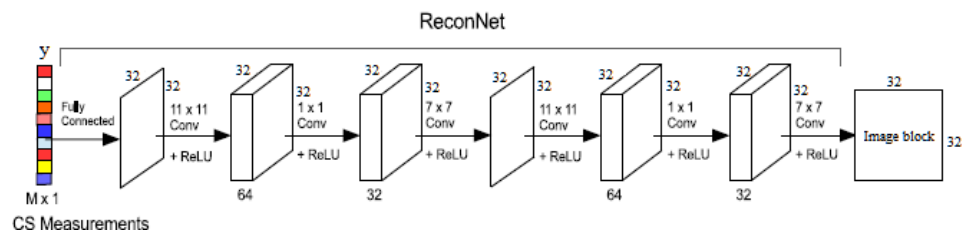


Figure 3.7: The network structure of ReconNet (Kulkarni et al., 2016).

In the network training process of this project, the number of epochs is set as 5000 and the batch size is set as 128. The mean squared error (MSE) is chosen as the loss function and adadelata as the optimizer. Early stopping is used with patience set as 100 and validation loss as the loss monitor. All network trainings were done on the Universiti Tunku Abdul Rahman research server with a Dual Socket Xeon Silver 4112 (Total 8 cores, 16 threads), total 64 GB of memory, and a Nvidia Quadro P4000 (GP104GL) graphic card. The operating system is Linux Ubuntu 18.04 LTS Server.

3.6 Performance Evaluations

In order to quantify the image quality of the recovered images, the image quality is measured with peak signal-to-noise ratio (PSNR) in dB, root mean

square error (RMSE), and structural similarity index measure (SSIM). Furthermore, the different image block sizes, SVR sensing matrices, the traditional CS image recovery algorithms and CNN are compared and determined their efficiency based on the time cost they required for image recovery.

In terms of mathematics, PSNR is defined as

$$PSNR = 20 \cdot \log_{10}(MAX_I) - 10 \cdot \log_{10}(MSE) \quad (3.10)$$

where MAX_I is the maximum pixel value of the image, and MSE is the mean square error defined as

$$MSE = \frac{1}{N} \sum_{i=1}^N (x_i - x'_i)^2 \quad (3.11)$$

where N is the number of image pixels, x_i and x'_i are the i -th image pixel of the original and recovered image, respectively. RMSE is simply the square root of Equation (3.11),

$$RMSE = \sqrt{MSE} = \sqrt{\frac{1}{N} \sum_{i=1}^N (x_i - x'_i)^2}, \quad (3.12)$$

and SSIM is defined as

$$SSIM = \frac{(2 \mu_x \mu_{x'} + C_1)(2 \sigma_{xx'} + C_2)}{(\mu_x^2 + \mu_{x'}^2 + C_1)(\sigma_x^2 + \sigma_{x'}^2 + C_2)}, \quad (3.13)$$

where μ_x is the mean of x , $\mu_{x'}$ is the means of x' , σ_x is the variance of x , $\sigma_{x'}$ is the variance of x' , $\sigma_{xx'}$ is the covariance of x and x' . $C_1 = (k_1 L)^2$, $C_2 = (k_2 L)^2$ are the two variables to stabilize the division with weak denominator, where L is the dynamic range of the pixel values and $k_1 = 0.01$, $k_2 = 0.03$ by default. Aside from image quality, the time cost for image recovery has also been factored in to determine the time efficiency.

CHAPTER 4

RESULTS AND DISCUSSION

4.1 Programmable Spatially Variant Resolution (SVR) Single-Pixel Imaging

4.1.1 Evaluation of CPA Sensing Matrices

As different values of z_0 generate different SVR CPA sensing matrices, it is worth to examine the effects of different values of z_0 on the image quality. In the numerical experiments, five values $z_0 \in \{0.19, 0.39, 0.59, 0.79, 0.99\}$ were used to generate different SVR CPA sensing matrices. Table 4.1 – 4.4 show the values of PSNR (in dB), RMSE, SSIM, and the time cost for image recovery (in seconds) of each individual image recovered via SVR CPA sensing matrices generated with $z_0 \in \{0.19, 0.39, 0.59, 0.79, 0.99\}$ from $SR \in \{0.5, 0.25, 0.1, 0.04\}$. Figure 4.1 – 4.4 show the mean values of PSNR (in dB), RMSE, SSIM, and time cost for image recovery (in seconds) of image recovered via SVR CPA sensing matrices generated with $z_0 \in \{0.19, 0.39, 0.59, 0.79, 0.99\}$ from $SR \in \{0.5, 0.25, 0.1, 0.04\}$, where SR is defined as $SR = M/N$. All images were recovered through total variant minimization by augmented Lagrangian and alternating direction algorithms (TVAL3) (Li et al., 2009). The codes available on the authors' website were used in this project and all parameters were kept at their default values.

Table 4.1: The values of PSNR (in dB) of each individual image recovered via SVR CPA sensing matrices generated with $z_0 \in \{0.19, 0.39, 0.59, 0.79, 0.99\}$ from $SR \in \{0.5, 0.25, 0.1, 0.04\}$. The highest value in each setting is in bold.

Images	z_0	PSNR (dB)			
		SR = 0.5	0.25	0.1	0.04
baboon	0.19	15.8798	16.0411	16.3944	16.6952
	0.39	16.2450	16.0198	16.5775	15.7225
	0.59	16.4609	16.2603	16.1966	15.4533
	0.79	16.3019	16.5064	17.1577	16.1278
	0.99	16.4004	16.7214	15.2307	16.0004
Barbara	0.19	16.6165	16.6197	16.7248	15.5099
	0.39	17.2300	17.1489	17.1071	15.7228
	0.59	17.1845	17.1168	17.0018	15.6508
	0.79	17.1493	17.0860	17.0491	16.2029
	0.99	17.1695	17.0216	16.7595	16.1175
cat	0.19	16.0710	16.0981	16.4391	16.4838
	0.39	16.4396	16.4022	16.6342	17.2483
	0.59	16.4557	17.4567	16.7351	17.3195
	0.79	17.5484	17.2274	16.7652	17.4728
	0.99	16.5199	16.2836	16.8680	17.3455
fruits	0.19	15.6206	15.4313	15.0414	13.9411
	0.39	15.9243	15.8085	15.5567	14.3976
	0.59	15.9267	15.8130	15.5282	14.4380
	0.79	16.0852	15.7437	15.4632	13.6610
	0.99	16.0942	15.8811	15.2647	14.3030
Lenna	0.19	17.3097	17.3927	17.1252	16.2220
	0.39	17.8719	17.8982	18.1877	17.0640
	0.59	17.9832	18.0147	17.9783	16.2966
	0.79	18.1500	18.2047	17.6708	16.7358
	0.99	18.0044	18.1144	18.0806	16.3854
peppers	0.19	16.1234	16.0247	15.9437	14.6426
	0.39	16.3409	16.4322	17.2350	14.0624
	0.59	16.3036	16.1318	16.5171	16.1648

	0.79	18.0241	16.7225	16.0872	15.9762
	0.99	16.3457	16.7115	16.0474	16.0037
airplane	0.19	17.5973	17.3666	16.4229	15.8513
	0.39	17.8679	17.9425	18.3390	15.0162
	0.59	17.6889	17.6233	17.7353	15.4903
	0.79	17.8879	18.0681	18.7311	16.6123
	0.99	17.9333	17.7780	17.0706	14.9369
cameraman	0.19	17.5280	17.4943	14.1790	14.3927
	0.39	19.7297	17.1167	17.2133	13.9136
	0.59	19.9100	18.8815	14.7577	16.5714
	0.79	19.5632	16.3292	15.9325	15.4482
	0.99	18.5027	17.7246	16.0187	13.1784
Zelda	0.19	13.2272	13.3127	14.3337	15.3757
	0.39	12.9585	12.8286	12.5640	12.0770
	0.59	13.1521	12.9372	13.9567	15.1492
	0.79	12.9012	12.9985	12.4059	13.0650
	0.99	12.9674	12.9457	12.8337	12.9483
Goldhill	0.19	19.8609	19.8413	19.6447	17.7451
	0.39	19.8472	19.6488	19.6661	16.9829
	0.59	19.8542	20.0305	19.8762	18.1514
	0.79	19.8873	19.9546	19.4622	19.7406
	0.99	19.8007	19.8568	19.4804	17.1019
Mean	0.19	16.5835	16.5622	16.2249	15.6859
	0.39	17.0455	16.7246	16.9081	15.2207
	0.59	17.0920	17.0266	16.6283	16.0685
	0.79	17.3499	16.8841	16.6725	16.1043
	0.99	16.9738	16.9039	16.3654	15.4321

Table 4.2: The values of RMSE of each individual image recovered via SVR CPA sensing matrices generated with $z_0 \in \{0.19, 0.39, 0.59, 0.79, 0.99\}$ from $SR \in \{0.5, 0.25, 0.1, 0.04\}$. The lowest value in each setting is in bold.

Images	z_0	RMSE			
		SR = 0.5	0.25	0.1	0.04
baboon	0.19	0.1607	0.1577	0.1515	0.1463
	0.39	0.1541	0.1581	0.1483	0.1636
	0.59	0.1503	0.1538	0.1549	0.1688
	0.79	0.1531	0.1495	0.1387	0.1562
	0.99	0.1513	0.1459	0.1732	0.1585
Barbara	0.19	0.1476	0.1476	0.1458	0.1677
	0.39	0.1376	0.1389	0.1395	0.1636
	0.59	0.1383	0.1394	0.1412	0.1650
	0.79	0.1388	0.1399	0.1405	0.1548
	0.99	0.1385	0.1409	0.1452	0.1564
cat	0.19	0.1572	0.1567	0.1507	0.1499
	0.39	0.1507	0.1513	0.1473	0.1373
	0.59	0.1504	0.1340	0.1456	0.1362
	0.79	0.1326	0.1376	0.1451	0.1338
	0.99	0.1493	0.1534	0.1434	0.1357
fruits	0.19	0.1656	0.1692	0.1770	0.2009
	0.39	0.1599	0.1620	0.1668	0.1906
	0.59	0.1598	0.1619	0.1673	0.1897
	0.79	0.1569	0.1632	0.1686	0.2075
	0.99	0.1568	0.1607	0.1725	0.1927
Lenna	0.19	0.1363	0.1350	0.1392	0.1545
	0.39	0.1278	0.1274	0.1232	0.1402
	0.59	0.1261	0.1257	0.1262	0.1532
	0.79	0.1237	0.1230	0.1308	0.1456
	0.99	0.1258	0.1242	0.1247	0.1516
peppers	0.19	0.1563	0.1580	0.1595	0.1853
	0.39	0.1524	0.1508	0.1375	0.1981
	0.59	0.1530	0.1561	0.1493	0.1555

	0.79	0.1255	0.1458	0.1569	0.1589
	0.99	0.1523	0.1460	0.1576	0.1584
airplane	0.19	0.1319	0.1354	0.1510	0.1612
	0.39	0.1278	0.1267	0.1211	0.1775
	0.59	0.1305	0.1315	0.1298	0.1681
	0.79	0.1275	0.1249	0.1157	0.1477
	0.99	0.1269	0.1292	0.1401	0.1791
cameraman	0.19	0.1329	0.1334	0.1955	0.1907
	0.39	0.1032	0.1394	0.1378	0.2015
	0.59	0.1010	0.1137	0.1829	0.1484
	0.79	0.1052	0.1526	0.1597	0.1689
	0.99	0.1188	0.1299	0.1581	0.2193
Zelda	0.19	0.2181	0.2160	0.1920	0.1703
	0.39	0.2249	0.2283	0.2354	0.2490
	0.59	0.2200	0.2255	0.2005	0.1748
	0.79	0.2264	0.2239	0.2397	0.2222
	0.99	0.2247	0.2253	0.2282	0.2252
Goldhill	0.19	0.1016	0.1018	0.1042	0.1296
	0.39	0.1018	0.1041	0.1039	0.1415
	0.59	0.1017	0.0996	0.1014	0.1237
	0.79	0.1013	0.1005	0.1064	0.1030
	0.99	0.1023	0.1017	0.1062	0.1396
Mean	0.19	0.1508	0.1511	0.1566	0.1656
	0.39	0.1440	0.1487	0.1461	0.1763
	0.59	0.1431	0.1441	0.1499	0.1583
	0.79	0.1391	0.1461	0.1502	0.1599
	0.99	0.1447	0.1457	0.1549	0.1717

Table 4.3: The values of SSIM of each individual image recovered via SVR CPA sensing matrices generated with $z_0 \in \{0.19, 0.39, 0.59, 0.79, 0.99\}$ from $SR \in \{0.5, 0.25, 0.1, 0.04\}$. The highest value in each setting is in bold.

Images	z_0	SSIM			
		SR = 0.5	0.25	0.1	0.04
baboon	0.19	0.4215	0.4073	0.3663	0.2735
	0.39	0.4402	0.4323	0.3851	0.2851
	0.59	0.4418	0.4306	0.3851	0.2861
	0.79	0.4434	0.4292	0.3917	0.2784
	0.99	0.4401	0.4337	0.3531	0.2869
Barbara	0.19	0.4581	0.4478	0.4059	0.2782
	0.39	0.5062	0.4968	0.4562	0.3091
	0.59	0.5059	0.4979	0.4666	0.2964
	0.79	0.5066	0.4999	0.4654	0.2970
	0.99	0.5058	0.4985	0.4572	0.3132
cat	0.19	0.4926	0.4730	0.4237	0.3372
	0.39	0.5834	0.5631	0.5047	0.3813
	0.59	0.5873	0.5744	0.5091	0.4036
	0.79	0.5905	0.5727	0.5081	0.3854
	0.99	0.5847	0.5667	0.4954	0.3898
fruits	0.19	0.4788	0.4646	0.3995	0.2399
	0.39	0.5047	0.4869	0.4282	0.2819
	0.59	0.5024	0.4823	0.4291	0.2729
	0.79	0.4997	0.4903	0.4294	0.2465
	0.99	0.4998	0.4745	0.4271	0.2640
Lenna	0.19	0.5107	0.5062	0.4762	0.3473
	0.39	0.5970	0.5870	0.5632	0.4229
	0.59	0.6002	0.5969	0.5550	0.3982
	0.79	0.6063	0.5981	0.5512	0.4173
	0.99	0.6014	0.5949	0.5656	0.3942
peppers	0.19	0.5675	0.5579	0.5022	0.3471
	0.39	0.5713	0.5629	0.5251	0.3519
	0.59	0.5702	0.5585	0.5128	0.3788

	0.79	0.5794	0.5643	0.5141	0.3969
	0.99	0.5706	0.5650	0.5094	0.3536
airplane	0.19	0.5951	0.5756	0.4909	0.3689
	0.39	0.6255	0.5989	0.5246	0.3522
	0.59	0.6275	0.6080	0.5316	0.3811
	0.79	0.6289	0.6132	0.5214	0.3918
	0.99	0.6279	0.6062	0.5281	0.3648
cameraman	0.19	0.5610	0.5560	0.4886	0.3681
	0.39	0.6497	0.6274	0.5735	0.4198
	0.59	0.6649	0.6385	0.5506	0.4249
	0.79	0.6528	0.6344	0.5624	0.4244
	0.99	0.6529	0.6349	0.5748	0.4271
Zelda	0.19	0.5162	0.5083	0.4732	0.3469
	0.39	0.5548	0.5499	0.5012	0.3362
	0.59	0.5585	0.5480	0.5117	0.3449
	0.79	0.5542	0.5489	0.4990	0.3435
	0.99	0.5555	0.5487	0.5040	0.3596
Goldhill	0.19	0.5380	0.5227	0.4728	0.3669
	0.39	0.5391	0.5223	0.4810	0.3584
	0.59	0.5405	0.5294	0.4644	0.3577
	0.79	0.5404	0.5285	0.4682	0.3811
	0.99	0.5406	0.5315	0.4685	0.3450
Mean	0.19	0.5139	0.5019	0.4499	0.3274
	0.39	0.5572	0.5427	0.4943	0.3499
	0.59	0.5599	0.5465	0.4916	0.3545
	0.79	0.5602	0.5480	0.4911	0.3562
	0.99	0.5580	0.5454	0.4883	0.3498

Table 4.4: Time cost for image recovery (in seconds) of each individual image recovered via SVR CPA sensing matrices generated with $z_0 \in \{0.19, 0.39, 0.59, 0.79, 0.99\}$ from $SR \in \{0.5, 0.25, 0.1, 0.04\}$. The lowest time cost in each setting is in bold.

Images	z_0	Time cost for image recovery (s)			
		SR = 0.5	0.25	0.1	0.04
baboon	0.19	1.78	1.05	0.58	0.45
	0.39	2.42	1.05	0.75	0.47
	0.59	2.06	1.06	0.69	0.44
	0.79	2.11	1.31	0.64	0.47
	0.99	2.03	1.25	0.33	0.44
Barbara	0.19	1.86	0.98	0.64	0.55
	0.39	2.20	1.45	0.73	0.52
	0.59	2.27	1.19	0.75	0.59
	0.79	2.33	1.33	0.80	0.47
	0.99	2.23	1.16	0.72	0.53
cat	0.19	1.80	1.19	0.64	0.52
	0.39	2.08	1.42	0.84	0.50
	0.59	2.02	0.95	0.66	0.47
	0.79	2.20	1.30	0.66	0.55
	0.99	2.13	1.38	0.59	0.50
fruits	0.19	1.80	1.11	0.69	0.56
	0.39	2.28	1.42	0.72	0.47
	0.59	2.28	1.06	0.83	0.48
	0.79	2.30	1.13	0.78	0.52
	0.99	2.33	1.03	0.73	0.53
Lenna	0.19	1.86	0.89	0.58	0.48
	0.39	2.19	1.11	0.75	0.52
	0.59	2.13	1.11	0.78	0.61
	0.79	1.95	1.30	0.63	0.52
	0.99	2.13	1.14	0.97	0.52
peppers	0.19	1.83	1.14	0.59	0.56
	0.39	2.11	1.16	0.66	0.55
	0.59	2.06	1.17	0.75	0.53

	0.79	2.20	1.11	0.63	0.48
	0.99	2.08	1.02	0.67	0.58
airplane	0.19	1.73	0.88	0.52	0.44
	0.39	1.81	0.89	0.58	0.38
	0.59	1.66	1.02	0.56	0.38
	0.79	1.80	0.94	0.72	0.36
	0.99	1.89	0.97	0.63	0.45
cameraman	0.19	1.86	1.25	0.61	0.48
	0.39	2.11	1.17	0.63	0.39
	0.59	2.00	1.13	0.80	0.50
	0.79	2.19	1.11	0.66	0.45
	0.99	2.00	1.06	0.72	0.52
Zelda	0.19	2.28	1.09	0.69	0.45
	0.39	1.86	0.97	0.69	0.53
	0.59	1.88	1.00	0.72	0.44
	0.79	1.73	1.16	0.66	0.55
	0.99	1.97	1.13	0.66	0.50
Goldhill	0.19	2.05	1.03	0.61	0.52
	0.39	1.66	1.00	0.67	0.48
	0.59	1.81	0.97	0.64	0.50
	0.79	1.83	1.06	0.63	0.52
	0.99	1.78	0.97	0.72	0.47
Mean	0.19	1.88	1.06	0.61	0.50
	0.39	2.07	1.16	0.70	0.48
	0.59	2.02	1.07	0.72	0.49
	0.79	2.06	1.17	0.68	0.49
	0.99	2.06	1.11	0.67	0.50

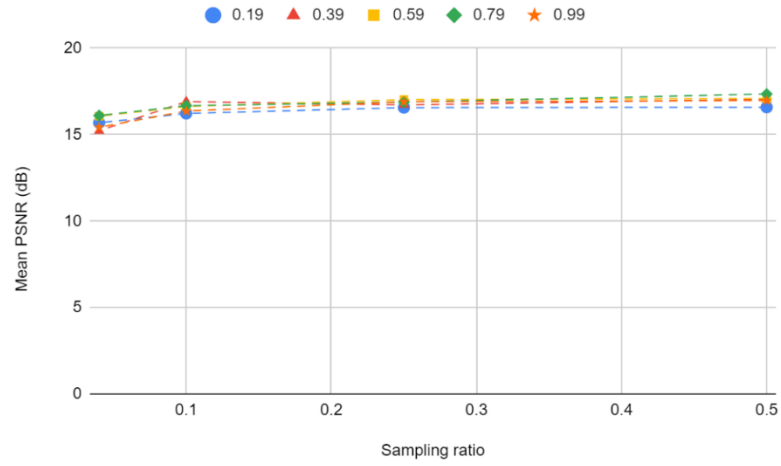


Figure 4.1: The mean values of PSNR (in dB) of image recovered via SVR CPA sensing matrices generated with $z_0 \in \{0.19, 0.39, 0.59, 0.79, 0.99\}$ from $SR \in \{0.5, 0.25, 0.1, 0.04\}$.

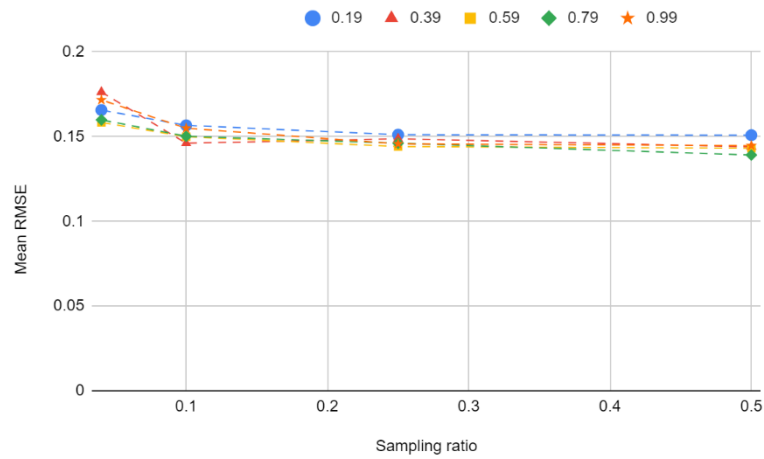


Figure 4.2: The mean values of RMSE of image recovered via SVR CPA sensing matrices generated with $z_0 \in \{0.19, 0.39, 0.59, 0.79, 0.99\}$ from $SR \in \{0.5, 0.25, 0.1, 0.04\}$.

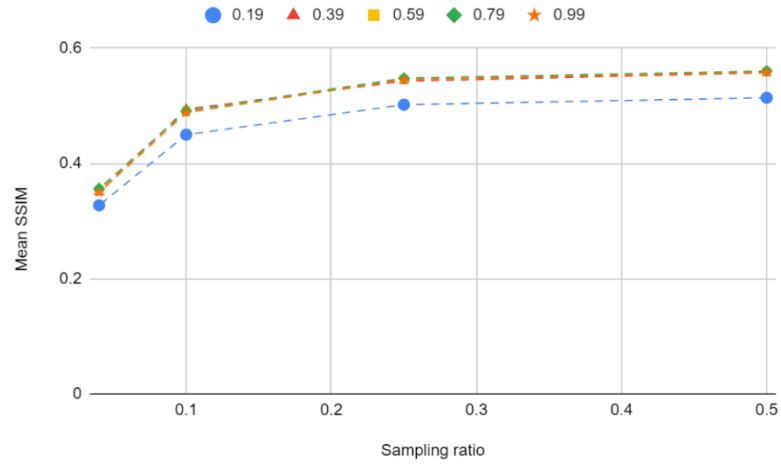


Figure 4.3: The mean values of SSIM of image recovered via SVR CPA sensing matrices generated with $z_0 \in \{0.19, 0.39, 0.59, 0.79, 0.99\}$ from $SR \in \{0.5, 0.25, 0.1, 0.04\}$.

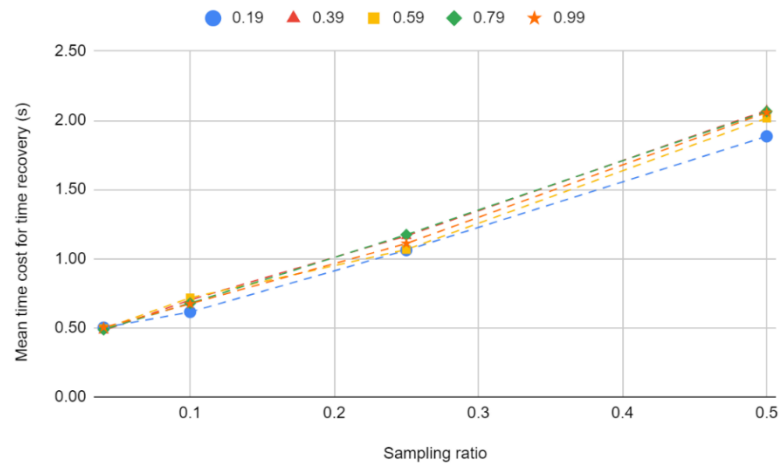


Figure 4.4: The mean time cost for image recovery (in seconds) of image recovered via SVR CPA sensing matrices generated with $z_0 \in \{0.19, 0.39, 0.59, 0.79, 0.99\}$ from $SR \in \{0.5, 0.25, 0.1, 0.04\}$.

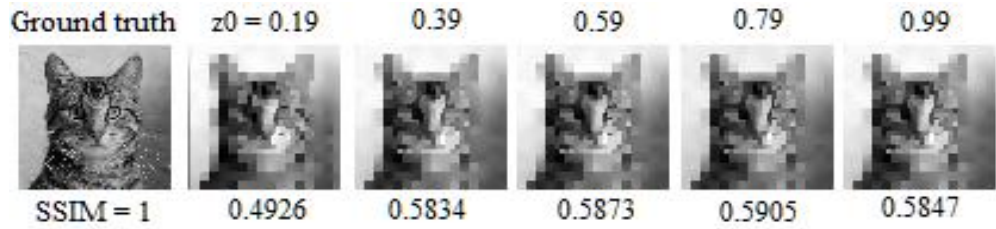


Figure 4.5: (Starting from left to right) The original image of “cat”, resized to 64×64 , and the images recovered via SVR CPA sensing matrices generated by $z_0 \in \{0.19, 0.39, 0.59, 0.79, 0.99\}$ from $SR = 0.5$.

From the results shown in Table 4.1 – 4.4 and Figure 4.1 – 4.4, it can be seen that the mean values of PSNR (in dB), RMSE, SSIM, and time cost for image recovery (in seconds) of images recovered via SVR CPA sensing matrices generated with $z_0 \in \{0.19, 0.39, 0.59, 0.79, 0.99\}$ from $SR \in \{0.5, 0.25, 0.1, 0.04\}$ do not differ drastically for different values of z_0 . In addition, Figure 4.5 shows that the image details of images recovered via SVR CPA sensing matrices generated by $z_0 \in \{0.19, 0.39, 0.59, 0.79, 0.99\}$ from $SR = 0.5$ do not make noticeable differences for different values of z_0 . Hence, it is acceptable to select any value of z_0 for the generation of SVR CPA sensing matrices. Specifically, this project uses $z_0 = 0.19$ to generate the SVR CPA sensing matrices.

4.1.2 Comparison of Various SVR Sensing Matrices

In this section, the performance of different SVR sensing matrices were compared extensively. The sensing matrices chosen for comparison are SVR RBP, HMP, and SVR sensing matrix. Table 4.5 – 4.8 show the values of PSNR (in dB), RMSE, SSIM, and time cost for image recovery (in seconds) of each

individual image recovered via SVR RBP, HMP, and CPA sensing matrices. Figure 4.6 – 4.9 show the mean values of PSNR (in dB), RMSE, SSIM, and time cost for image recovery (in seconds) of the images recovered via SVR RBP, HMP, and CPA sensing matrices.

Table 4.5: The values of PSNR (in dB) of each individual image recovered via SVR RBP, HMP, and CPA sensing matrices. The highest value in each setting is in bold.

Images	Sensing matrices	PSNR (dB)			
		SR = 0.5	0.25	0.1	0.04
baboon	RBP	15.2765	15.1805	14.8149	13.6834
	HMP	16.4932	16.8487	17.3453	15.5875
	CPA	15.8798	16.0411	16.3944	16.6952
Barbara	RBP	16.1633	15.9937	15.4689	15.1489
	HMP	17.5654	16.8602	16.4583	14.9381
	CPA	16.6165	16.6197	16.7248	15.5099
cat	RBP	15.6788	15.9398	15.5935	15.2511
	HMP	18.3728	16.8735	17.1940	13.4215
	CPA	16.0710	16.0981	16.4391	16.4838
fruits	RBP	14.6076	14.4842	13.2702	12.8509
	HMP	15.5036	15.5314	15.0953	12.4651
	CPA	15.6206	15.4313	15.0414	13.9411
Lenna	RBP	16.7298	16.2832	15.9699	14.4754
	HMP	19.2254	18.9017	17.2400	16.4930
	CPA	17.3097	17.3927	17.1252	16.2220
peppers	RBP	15.3167	16.9184	15.0797	15.0339
	HMP	16.6354	16.3577	15.8392	16.2259
	CPA	16.1234	16.0247	15.9437	14.6426
airplane	RBP	15.7174	15.4976	15.9087	15.4772
	HMP	18.1867	16.8344	17.4982	15.5117
	CPA	17.5973	17.3666	16.4229	15.8513
cameraman	RBP	13.4797	14.0262	11.8637	13.8225

	HMP	15.5045	14.8719	15.0279	12.2226
	CPA	17.5280	17.4943	14.1790	14.3927
Zelda	RBP	11.9627	11.7226	12.3820	13.0845
	HMP	14.2635	13.0302	13.4685	15.1228
	CPA	13.2272	13.3127	14.3337	15.3757
Goldhill	RBP	17.3198	17.4608	16.5779	15.7688
	HMP	19.0623	17.6219	17.8692	16.7405
	CPA	19.8609	19.8413	19.6447	17.7451
Mean	RBP	15.2252	15.3507	14.6929	14.4597
	HMP	17.0813	16.3731	16.3036	14.8729
	CPA	16.5835	16.5622	16.2249	15.6859

Table 4.6: The values of RMSE of each individual image recovered via SVR RBP, HMP, and CPA sensing matrices from $SR \in \{0.5, 0.25, 0.1, 0.04\}$. The lowest value in each setting is in bold.

Images	Sensing matrices	RMSE			
		SR = 0.5	0.25	0.1	0.04
baboon	RBP	0.1723	0.1742	0.1817	0.2069
	HMP	0.1497	0.1437	0.1357	0.1662
	CPA	0.1607	0.1577	0.1515	0.1463
Barbara	RBP	0.1555	0.1586	0.1685	0.1748
	HMP	0.1324	0.1435	0.1503	0.1791
	CPA	0.1476	0.1476	0.1458	0.1677
cat	RBP	0.1645	0.1596	0.1661	0.1728
	HMP	0.1206	0.1433	0.1381	0.2133
	CPA	0.1572	0.1567	0.1507	0.1499
fruits	RBP	0.1860	0.1887	0.2170	0.2277
	HMP	0.1678	0.1673	0.1759	0.2381
	CPA	0.1656	0.1692	0.1770	0.2009
Lenna	RBP	0.1457	0.1534	0.1590	0.1889
	HMP	0.1093	0.1135	0.1374	0.1497
	CPA	0.1363	0.1350	0.1392	0.1545
peppers	RBP	0.1715	0.1426	0.1762	0.1771
	HMP	0.1473	0.1521	0.1615	0.1544

	CPA	0.1563	0.1580	0.1595	0.1853
airplane	RBP	0.1637	0.1679	0.1602	0.1683
	HMP	0.1232	0.1440	0.1334	0.1677
	CPA	0.1319	0.1354	0.1510	0.1612
cameraman	RBP	0.2118	0.1989	0.2552	0.2387
	HMP	0.1678	0.1805	0.1773	0.2012
	CPA	0.1329	0.1334	0.1955	0.1907
Zelda	RBP	0.2523	0.2593	0.2404	0.2217
	HMP	0.1936	0.2231	0.2121	0.1753
	CPA	0.2181	0.2160	0.1920	0.1703
Goldhill	RBP	0.1361	0.1340	0.1483	0.1628
	HMP	0.1114	0.1315	0.1278	0.1455
	CPA	0.1016	0.1018	0.1042	0.1296
Mean	RBP	0.1759	0.1737	0.1872	0.1940
	HMP	0.1423	0.1542	0.1550	0.1791
	CPA	0.1508	0.1511	0.1566	0.1656

Table 4.7: The values of SSIM of each individual image recovered via SVR RBP, HMP, and CPA sensing matrices from $SR \in \{0.5, 0.25, 0.1, 0.04\}$. The highest value in each setting is in bold.

Images	Sensing matrices	SSIM			
		SR = 0.5	0.25	0.1	0.04
baboon	RBP	0.2937	0.3016	0.2450	0.2237
	HMP	0.3974	0.4007	0.3531	0.2511
	CPA	0.4215	0.4073	0.3663	0.2735
Barbara	RBP	0.3457	0.3197	0.2663	0.2120
	HMP	0.4877	0.4511	0.3865	0.3097
	CPA	0.4581	0.4478	0.4059	0.2782
cat	RBP	0.3603	0.3461	0.3310	0.2664
	HMP	0.4491	0.4649	0.3766	0.2457
	CPA	0.4926	0.4730	0.4237	0.3372
fruits	RBP	0.3339	0.3171	0.2849	0.1744
	HMP	0.4646	0.4565	0.3777	0.2165
	CPA	0.4788	0.4646	0.3995	0.2399

Lenna	RBP	0.3919	0.3379	0.3143	0.2444
	HMP	0.5590	0.5330	0.4507	0.3702
	CPA	0.5107	0.5062	0.4762	0.3473
peppers	RBP	0.4439	0.4285	0.3705	0.3166
	HMP	0.5831	0.5608	0.4979	0.3915
	CPA	0.5675	0.5579	0.5022	0.3471
airplane	RBP	0.4581	0.4471	0.4094	0.3762
	HMP	0.5713	0.5271	0.4728	0.3543
	CPA	0.5951	0.5756	0.4909	0.3689
cameraman	RBP	0.4234	0.4126	0.4113	0.3836
	HMP	0.5510	0.5223	0.4391	0.3374
	CPA	0.5610	0.5560	0.4886	0.3681
Zelda	RBP	0.4235	0.4037	0.3466	0.2773
	HMP	0.5917	0.5533	0.4958	0.3767
	CPA	0.5162	0.5083	0.4732	0.3469
Goldhill	RBP	0.3174	0.2959	0.2523	0.2131
	HMP	0.4592	0.4149	0.3788	0.2864
	CPA	0.5380	0.5227	0.4728	0.3669
Mean	RBP	0.3792	0.3610	0.3232	0.2688
	HMP	0.5114	0.4885	0.4229	0.3140
	CPA	0.5139	0.5019	0.4499	0.3274

Table 4.8: The time cost for image recovery (in seconds) of each individual image recovered via SVR RBP, HMP, and CPA sensing matrices from $SR \in \{0.5, 0.25, 0.1, 0.04\}$. The lowest time cost in each setting is in bold.

Images	Sensing matrices	Time cost for image recovery (s)			
		SR = 0.5	0.25	0.1	0.04
baboon	RBP	1.55	0.92	0.55	0.30
	HMP	1.47	0.69	0.28	0.69
	CPA	1.78	1.05	0.58	0.45
Barbara	RBP	1.63	1.03	0.61	0.41
	HMP	0.72	1.00	0.69	0.39

	CPA	1.86	0.98	0.64	0.55
cat	RBP	1.89	1.30	0.78	0.50
	HMP	0.63	0.91	0.53	0.78
	CPA	1.80	1.19	0.64	0.52
fruits	RBP	2.41	1.36	0.91	0.47
	HMP	1.67	1.03	0.78	0.55
	CPA	1.80	1.11	0.69	0.56
Lenna	RBP	1.84	0.94	0.66	0.36
	HMP	0.64	0.41	0.81	0.48
	CPA	1.86	0.89	0.58	0.48
peppers	RBP	1.95	1.09	0.61	0.47
	HMP	1.92	0.36	0.59	0.59
	CPA	1.83	1.14	0.59	0.56
airplane	RBP	1.48	0.94	0.45	0.28
	HMP	0.69	0.92	0.56	0.69
	CPA	1.73	0.88	0.52	0.44
cameraman	RBP	2.06	1.20	0.72	0.41
	HMP	1.64	1.02	0.77	0.53
	CPA	1.86	1.25	0.61	0.48
Zelda	RBP	1.95	1.20	0.66	0.38
	HMP	0.73	0.86	0.89	0.66
	CPA	2.28	1.09	0.69	0.45
Goldhill	RBP	2.03	1.09	0.64	0.39
	HMP	1.41	1.23	0.77	0.59
	CPA	2.05	1.03	0.61	0.52
Mean	RBP	1.88	1.11	0.66	0.40
	HMP	1.15	0.84	0.67	0.60
	CPA	1.88	1.06	0.61	0.50

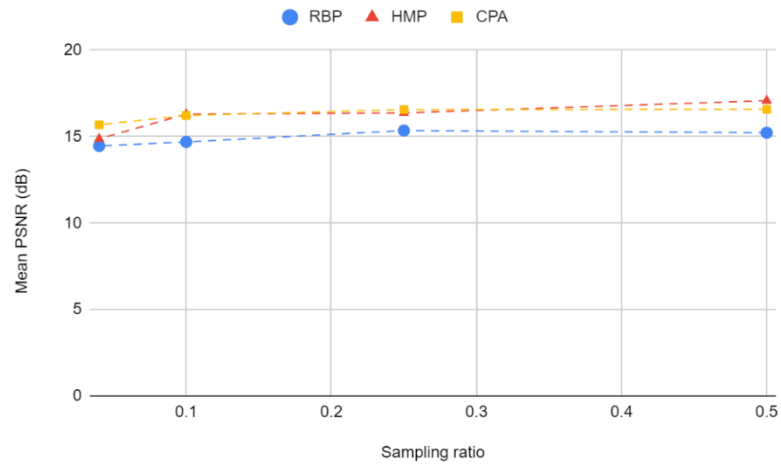


Figure 4.6: The mean values of PSNR (in dB) of images recovered via SVR RBP, HMP, and CPA sensing matrices from $SR \in \{0.5, 0.25, 0.1, 0.04\}$.

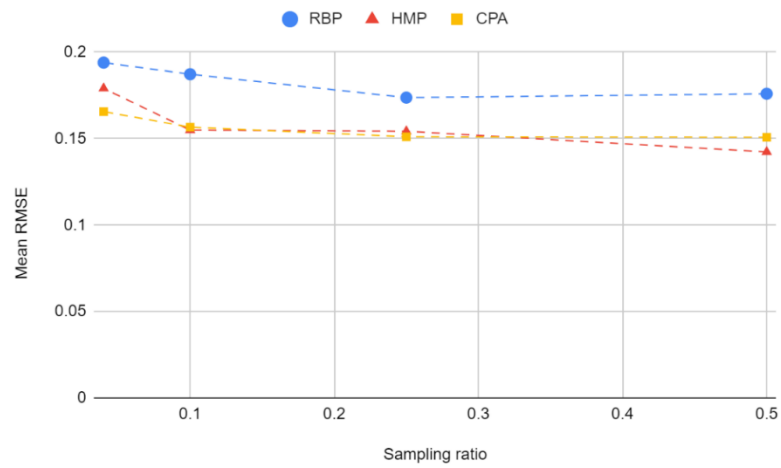


Figure 4.7: The mean values of RMSE of images recovered via SVR RBP, HMP, and CPA sensing matrices from $SR \in \{0.5, 0.25, 0.1, 0.04\}$.

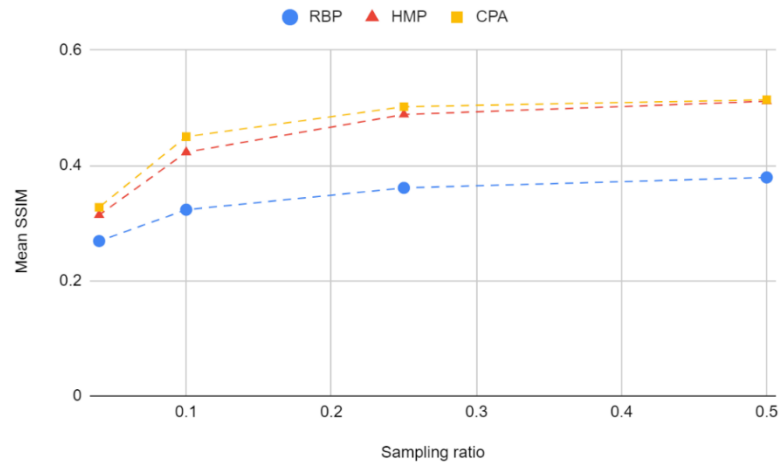


Figure 4.8: The mean values of SSIM of images recovered via SVR RBP, HMP, and CPA sensing matrices from $SR \in \{0.5, 0.25, 0.1, 0.04\}$.

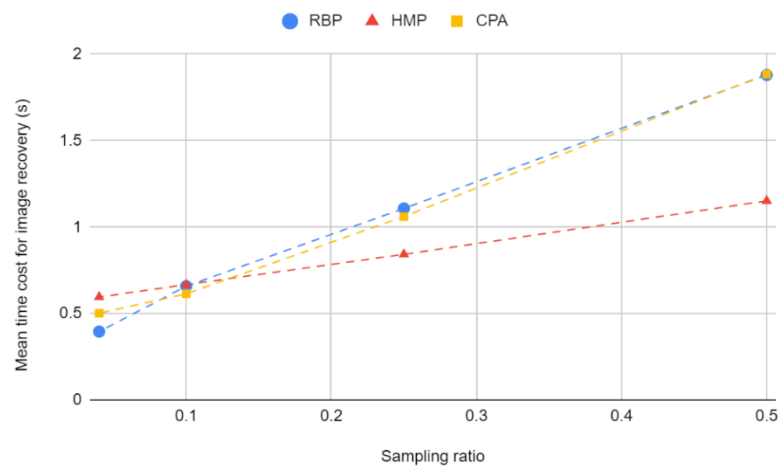


Figure 4.9: The mean time cost for image recovery (in seconds) of images recovered via SVR RBP, HMP, and CPA sensing matrices from $SR \in \{0.5, 0.25, 0.1, 0.04\}$.

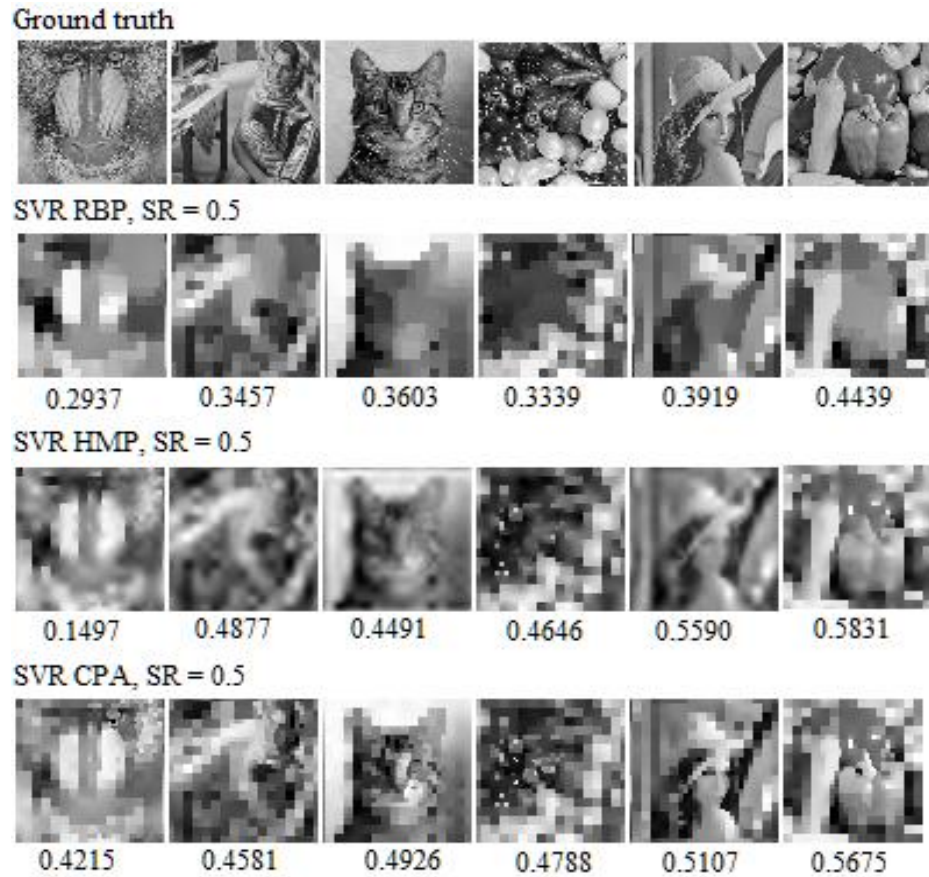


Figure 4.10: (First row) The original images of size 64×64 . (Second row) Images recovered via SVR RBP sensing matrix from $SR = 0.5$ with their respective SSIM. (Third row) Images recovered via SVR HMP sensing matrix from $SR = 0.5$ with their respective SSIM. (Fourth row) Images recovered via SVR CPA sensing matrix from $SR = 0.5$ with their respective SSIM.

According to the results shown in Table 4.5 – 4.8 and Figure 4.6 – 4.9, the values of the mean PSNR of images recovered via SVR CPA and HMP sensing matrices are comparable and higher than SVR RBP sensing matrices for most of the values of SR tested. The values of the mean RMSE of images recovered via SVR CPA and HMP sensing matrices are comparable and lower than SVR RBP sensing matrices for most of the values of SR tested. The mean

SSIM of images recovered via SVR CPA is in general slightly higher than that of images recovered via SVR HMP sensing matrices and significantly higher than that of images recovered via SVR RBP sensing matrices for all values of SR tested. In addition, the mean times cost for image recovery of images recovered via SVR HMP sensing matrices is in general greatly lower than that of images recovered via SVR RBP and CPA sensing matrices for $SR = 0.5, 0.25, \text{ and } 0.1$.

Although the values of the mean of PSNR, RMSE, and SSIM indicate that SVR CPA and HMP sensing matrices give a comparable image quality, Figure 4.10 shows that visually the image details in the fovea of images recovered via SVR CPA sensing matrices are much clearer than that of images recovered via SVR HMP sensing matrices. Therefore, for the trade-off of time cost for image recovery, CPA is more suitable for SVR approach than RBP and HMP.

4.1.3 Comparison of UR and SVR CPA Sensing Matrices

To begin with, the image quality of the images recovered via UR and SVR sensing matrices are compared extensively. For this experiment, all of the testing images were resized to 64×64 to accommodate the limited computational resources. Figure 4.11 shows the workflow of single-pixel imaging with CS via UR and SVR approaches in this project. Figure 4.12 shows a demonstration of how the fovea can be programmed to prioritize different regions of the images. For the sake of convenience, this project makes use of the built-in function *imtranslate* to swift the positions of the fovea in the images. Table 4.9 – 4.12 show the values of PSNR (in dB), RMSE, SSIM, and time cost

for image recovery (in seconds) for images recovered via UR and SVR CPA sensing matrices from $SR \in \{0.5, 0.25, 0.1, 0.04\}$. Figure 4.13 – 4.16 show the mean values of PSNR (in dB), RMSE, SSIM, and time cost for image recovery (in seconds) for images recovered via SVR CPA sensing matrices from $SR \in \{0.5, 0.25, 0.1, 0.04\}$.

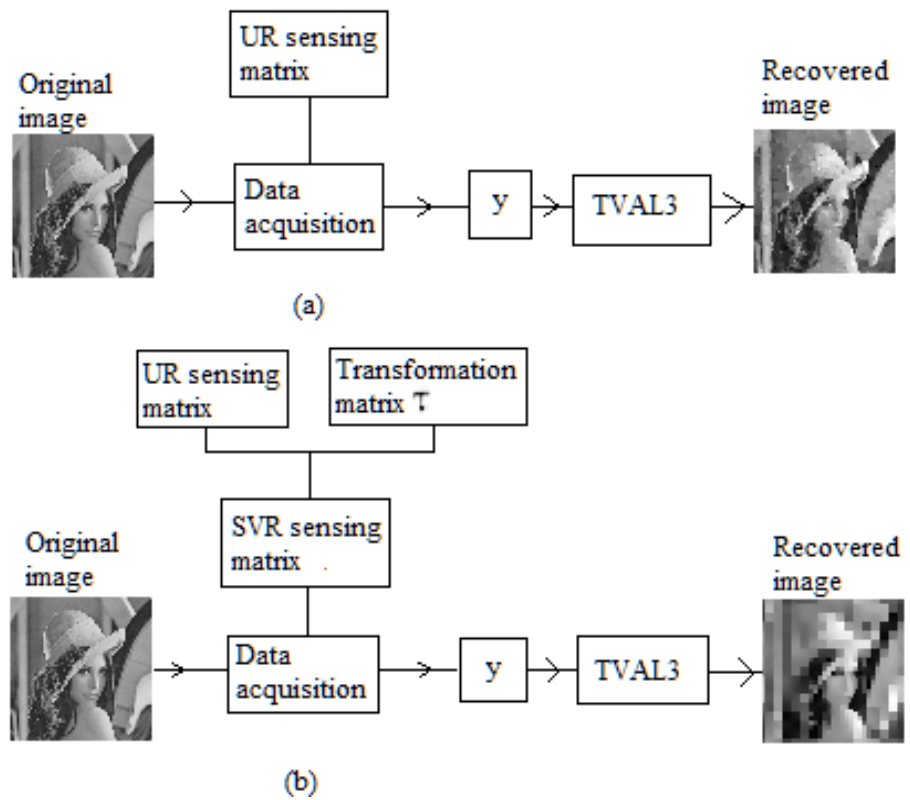


Figure 4.11: Workflow of single-pixel imaging with CS via (a) UR and (b) SVR approaches in this project.



Figure 4.12: (From left to right) A demonstration of programming the fovea to prioritize different regions of “cat”. Notice that except for the fovea, the rest of the regions of the image is in low-resolution.

Table 4.9: The values of PSNR (in dB) of each individual image recovered via UR and SVR CPA sensing matrices from $SR \in \{0.5, 0.25, 0.1, 0.04\}$. The highest value in each setting are in bold.

Images	Resolution	PSNR (dB)			
		SR = 0.5	0.25	0.1	0.04
baboon	UR	20.3870	17.9723	15.1414	15.2254
	SVR	15.8798	16.0411	16.3944	16.6952
Barbara	UR	20.7637	18.5377	14.9195	15.0081
	SVR	16.6165	16.6197	16.7248	15.5099
cat	UR	21.3825	20.2748	17.2828	16.4901
	SVR	16.0710	16.0981	16.4391	16.4838
fruits	UR	17.9164	16.2035	13.2548	10.9012
	SVR	15.6206	15.4313	15.0414	13.9411
Lenna	UR	23.5226	21.3353	18.3045	16.2273
	SVR	17.3097	17.3927	17.1252	16.2220
peppers	UR	22.5987	21.1266	17.6074	15.3072
	SVR	16.1234	16.0247	15.9437	14.6426
airplane	UR	25.9761	21.6107	12.7297	16.0656
	SVR	17.5973	17.3666	16.4229	15.8513
cameraman	UR	28.9251	22.5000	16.8575	13.2955
	SVR	17.5280	17.4943	14.1790	14.3927
Zelda	UR	18.6116	15.8081	18.2846	16.4490
	SVR	13.2272	13.3127	14.3337	15.3757
Goldhill	UR	24.6363	21.6251	18.6856	17.1611

	SVR	19.8609	19.8413	19.6447	17.7451
Mean	UR	22.4720	19.6994	16.3068	15.2130
	SVR	16.5835	16.5622	16.2249	15.6859

Table 4.10: The values of RMSE of each individual image recovered via UR and SVR CPA sensing matrices from $SR \in \{0.5, 0.25, 0.1, 0.04\}$ The lowest value in each setting is in bold.

Images	Resolution	RMSE			
		SR = 0.5	0.25	0.1	0.04
baboon	UR	0.0956	0.1263	0.1750	0.1733
	SVR	0.1607	0.1577	0.1515	0.1463
Barbara	UR	0.0916	0.1183	0.1795	0.1777
	SVR	0.1476	0.1476	0.1458	0.1677
cat	UR	0.0853	0.0961	0.1367	0.1498
	SVR	0.1572	0.1567	0.1507	0.1499
fruits	UR	0.1271	0.1548	0.2174	0.2851
	SVR	0.1656	0.1692	0.1770	0.2009
Lenna	UR	0.0667	0.0858	0.1216	0.1544
	SVR	0.1363	0.1350	0.1392	0.1545
peppers	UR	0.0741	0.0878	0.1317	0.1716
	SVR	0.1563	0.1580	0.1595	0.1853
airplane	UR	0.0503	0.0831	0.2309	0.1573
	SVR	0.1319	0.1354	0.1510	0.1612
cameraman	UR	0.0358	0.0750	0.1436	0.2164
	SVR	0.1329	0.1334	0.1955	0.1907
Zelda	UR	0.1173	0.1620	0.1218	0.1505
	SVR	0.2181	0.2160	0.1920	0.1703
Goldhill	UR	0.0586	0.0829	0.1163	0.1387
	SVR	0.1016	0.1018	0.1042	0.1296
Mean	UR	0.0802	0.1072	0.1575	0.1775
	SVR	0.1508	0.1511	0.1566	0.1656

Table 4.11: The values of SSIM of each individual image recovered via UR and SVR CPA sensing matrices from $SR \in \{0.5, 0.25, 0.1, 0.04\}$. The highest value in each setting are in bold.

Images	Resolution	SSIM			
		SR = 0.5	0.25	0.1	0.04
baboon	UR	0.6312	0.4327	0.2553	0.1836
	SVR	0.4215	0.4073	0.3663	0.2735
Barbara	UR	0.7701	0.6045	0.2980	0.2373
	SVR	0.4581	0.4478	0.4059	0.2782
cat	UR	0.7306	0.5629	0.3801	0.2655
	SVR	0.4926	0.4730	0.4237	0.3372
fruits	UR	0.7665	0.5716	0.3096	0.1610
	SVR	0.4788	0.4646	0.3995	0.2399
Lenna	UR	0.8492	0.5975	0.4584	0.2700
	SVR	0.5107	0.5062	0.4762	0.3473
peppers	UR	0.8630	0.7097	0.4677	0.2639
	SVR	0.5675	0.5579	0.5022	0.3471
airplane	UR	0.8307	0.6553	0.4153	0.3312
	SVR	0.5951	0.5756	0.4909	0.3689
cameraman	UR	0.8705	0.6879	0.4344	0.3532
	SVR	0.5610	0.5560	0.4886	0.3681
Zelda	UR	0.7998	0.6579	0.4722	0.2771
	SVR	0.5162	0.5083	0.4732	0.3469
Goldhill	UR	0.7665	0.5881	0.3974	0.2624
	SVR	0.5380	0.5227	0.4728	0.3669
Mean	UR	0.7878	0.6068	0.3888	0.2605
	SVR	0.5139	0.5019	0.4499	0.3274

Table 4.12: The time cost for image recovery of each individual image recovered via UR and SVR CPA sensing matrices from $SR \in \{0.5, 0.25, 0.1, 0.04\}$. The lowest time cost in each setting is in bold.

Images	Resolution	Time cost for image recovery (s)			
		SR = 0.5	0.25	0.1	0.04
baboon	UR	2.20	1.33	0.80	0.70
	SVR	1.78	1.05	0.58	0.45
Barbara	UR	2.05	1.19	0.75	0.48
	SVR	1.86	0.98	0.64	0.55
cat	UR	1.97	1.23	0.73	0.63
	SVR	1.80	1.19	0.64	0.52
fruits	UR	2.17	1.34	0.97	0.63
	SVR	1.80	1.11	0.69	0.56
Lenna	UR	2.08	1.59	0.83	0.66
	SVR	1.86	0.89	0.58	0.48
peppers	UR	1.95	1.31	1.03	0.56
	SVR	1.83	1.14	0.59	0.56
airplane	UR	2.48	1.47	0.86	0.56
	SVR	1.73	0.88	0.52	0.44
cameraman	UR	2.06	1.30	1.05	0.63
	SVR	1.86	1.25	0.61	0.48
Zelda	UR	2.09	1.31	0.88	0.66
	SVR	2.28	1.09	0.69	0.45
Goldhill	UR	2.02	1.27	0.78	0.64
	SVR	2.05	1.03	0.61	0.52
Mean	UR	2.11	1.33	0.87	0.61
	SVR	1.88	1.06	0.61	0.50

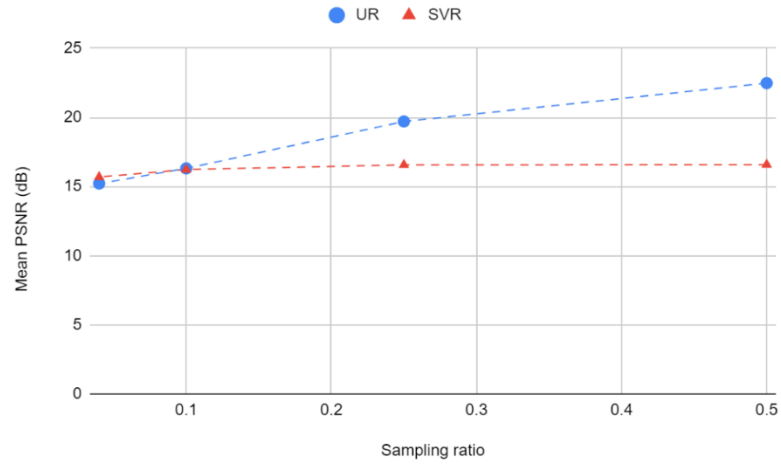


Figure 4.13: The mean values of PSNR (in dB) of images recovered via UR and SVR CPA sensing matrices from $SR \in \{0.5, 0.25, 0.1, 0.04\}$.

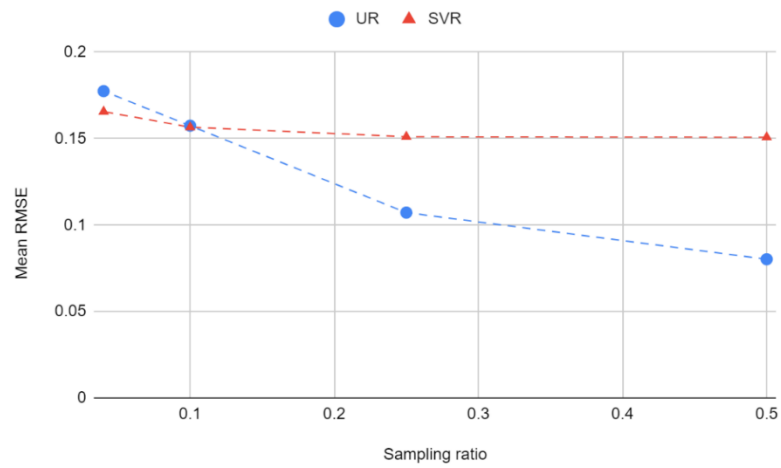


Figure 4.14: The mean values of RMSE of images recovered via UR and SVR CPA sensing matrices from $SR \in \{0.5, 0.25, 0.1, 0.04\}$.

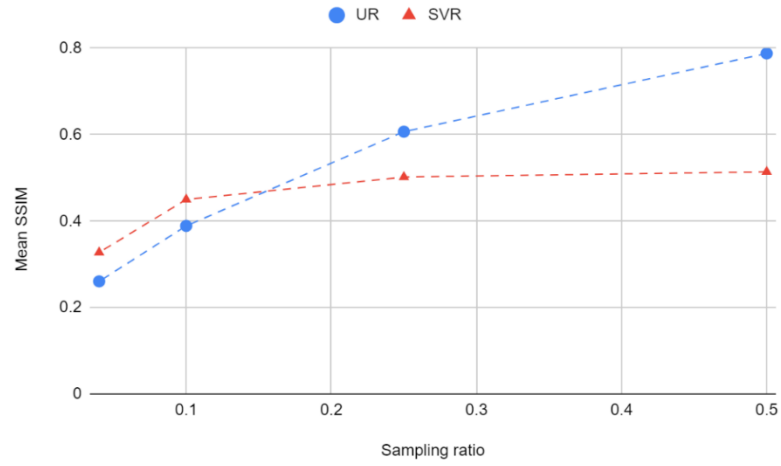


Figure 4.15: The mean values of SSIM of images recovered via UR and SVR CPA sensing matrices from $SR \in \{0.5, 0.25, 0.1, 0.04\}$.

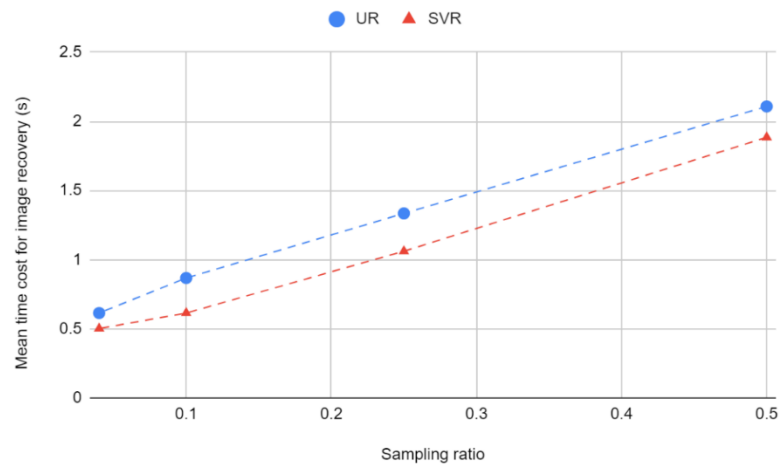
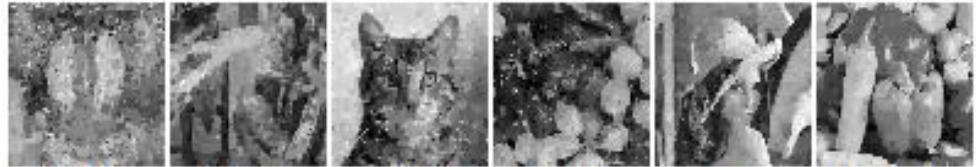


Figure 4.16: The mean time cost for image recovery (in seconds) of images recovered via UR and SVR CPA sensing matrices from $SR \in \{0.5, 0.25, 0.1, 0.04\}$.

Ground truth



UR, SR = 0.5



0.6312

0.7701

0.7306

0.7665

0.8492

0.8630

SVR, SR = 0.5



0.4215

0.4581

0.4926

0.4788

0.5107

0.5675

Figure 4.17: (First row) The original images of size 64×64 . (Second row) Images recovered via UR CPA sensing matrix from $SR = 0.5$ with their SSIM. (Third row) Images recovered via SVR CPA sensing matrix from $SR = 0.5$ with their respective values of SSIM.

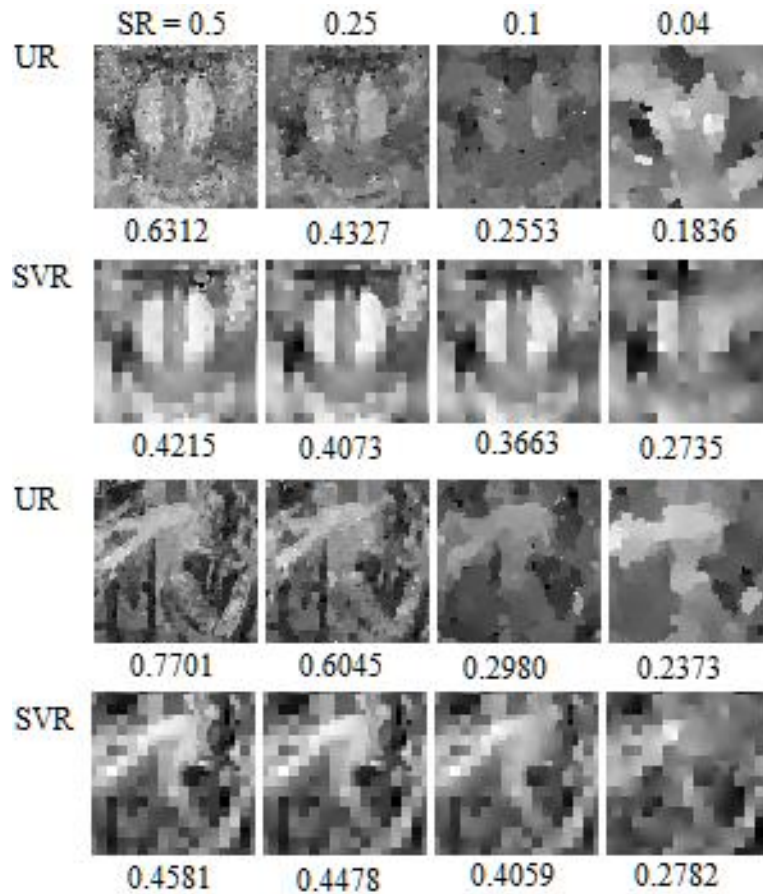


Figure 4.18: “baboon” and “Barbara” recovered via UR and SVR CPA sensing matrices from $SR \in \{0.5, 0.25, 0.1, 0.04\}$, shown with their respective values of SSIM.

According to the results shown in Table 4.9 – 4.12 and Figure 4.13 – 4.16, it can be seen that in general for images recovered via SVR CPA sensing matrices, the values of mean PSNR are higher for $SR = 0.04$, the values of mean RMSE are lower for $SR = 0.04$, and the values of mean SSIM are higher for $SR = 0.1$ and 0.04 . In addition, in general the time cost needed for image recovery of images recovered via SVR CPA sensing matrices are lower than that of images recovered via UR CPA sensing matrices throughout the range of SR tested in the numerical experiments.

Figure 4.17 shows the original images resized down to 64×64 , the images recovered via UR CPA sensing matrix from $SR = 0.5$ with their respective values of SSIM, and the images recovered via SVR CPA sensing matrix from $SR = 0.5$ with their respective values of SSIM. It shows that for the same SR , even though the values of SSIM of images recovered via UR CPA sensing matrices are higher than that of images recovered via SVR CPA sensing matrices, the SR required is high and not desirable for practical cases. In addition, Figure 4.18 shows “baboon” and “Barbara” recovered via UR and SVR CPA sensing matrices from $SR \in \{0.5, 0.25, 0.1, 0.04\}$, with their respective values of SSIM. It shows that as the SR gets lower, the quality of image recovered via UR CPA sensing matrices degrades much faster and more than the quality of image recovered via SVR CPA sensing matrices.

The results indicate that for lower values of SR , SVR CPA sensing matrices are able to improve the image quality. Therefore, SVR CPA sensing matrices are more suitable than UR CPA sensing matrices in situations where the numbers of measurements are small. Moreover, SVR CPA sensing matrices are also able to improve the time efficiency. Hence, SVR CPA sensing matrices are more favorable in practical cases since the time cost needed for image recovery is lower.

4.2 Block-Based SVR Single-Pixel Imaging with CS

4.2.1 Analysis on The Effects of Image Block Size

Intuitively, larger image block sizes require longer image processing time but there are fewer image blocks to be processed. On the other hand, smaller image

block sizes require shorter image processing time but there are more image blocks to be processed. Hence it is worthwhile to examine the influences of the size of image blocks has on the image quality of the recovered images. Naturally, the objective is to find an image block size that gives a higher image quality and requires less time cost for image recovery. For the numerical experiments, images were divided into several image block sizes $B \in \{64, 32, 16\}$. Same as in Section 4.1, the image blocks were recovered via TVAL3. Table 4.13 – 4.16 show the mean PSNR (in dB), RMSE, SSIM, and time cost for image recovery (in seconds) of the images formed from the image blocks with image block sizes $B \in \{64, 32, 16\}$ recovered via SVR CPA sensing matrices. Figure 4.19 – 4.22 show the values of mean PSNR, RMSE, SSIM, and time cost for image recovery (in seconds) of the images with image block size $B \in \{64, 32, 16\}$ recovered via SVR CPA sensing matrices. Figure 4.23 shows the ground truth of the testing images "baboon", "Barbara", "cat", "fruits", "Lena", "peppers", and the images formed from the recovered image blocks of size 16×16 , 32×32 , and 64×64 .

Table 4.13: The values of PSNR (in dB) of each individual image with image block size $B \in \{64, 32, 16\}$ recovered via SVR CPA sensing matrices from $SR \in \{0.25, 0.1, 0.04, 0.01\}$, with and without applying BM3D. The highest value in each setting is in bold.

Images	B	PSNR (dB)							
		SR = 0.25		0.1		0.04		0.01	
		w/o BM3D	w/ BM3D	w/o BM3D	w/ BM3D	w/o BM3D	w/ BM3D	w/o BM3D	w/ BM3D
baboon	64	18.2932	18.3568	20.6034	20.5720	18.1854	18.2502	18.1620	18.2269
	32	19.5647	19.1810	20.2633	20.1073	18.7481	18.8346	18.0761	18.2049
	16	18.8960	18.9547	17.7074	17.7349	19.3862	19.4357	13.7662	13.8424
Barbara	64	22.3329	22.4072	22.5115	22.6380	20.9105	21.0210	17.3444	17.4102
	32	15.7049	15.6334	16.9564	16.9383	21.1707	21.4387	18.1065	18.2605
	16	22.1686	22.3509	21.9790	22.2458	21.2242	21.6538	13.0603	13.1149
cat	64	19.0830	19.1376	19.1806	19.2432	21.3312	21.4770	19.4626	19.5755
	32	24.7117	24.4886	19.8213	19.7884	21.0090	21.1093	19.2642	19.4370
	16	19.5150	19.5977	22.8847	23.0284	21.9585	22.2526	12.1828	12.2039
fruits	64	23.6031	23.9278	23.0660	23.3430	20.8091	21.0138	15.2255	15.2739
	32	24.8748	24.9569	23.0326	23.2481	13.5098	13.4948	15.2710	15.3585
	16	23.1318	23.4847	22.8203	23.1868	15.3606	15.4073	10.0039	10.0196
Lenna	64	24.0907	24.4325	24.5612	24.9359	23.5503	23.8604	20.1212	20.2930
	32	25.0091	25.4457	23.5334	23.7916	23.0167	23.3666	20.0128	20.2447
	16	23.7610	24.2089	26.2523	26.9587	23.4572	24.1270	10.3892	10.4001
peppers	64	23.4226	23.6607	21.7691	21.9414	22.8360	23.1352	19.4874	19.6501
	32	22.4378	22.6201	22.0704	22.3131	24.0443	24.5079	19.1013	19.3142
	16	23.3282	23.6956	24.2038	24.7105	22.0382	22.5707	9.4925	9.5000
airplane	64	22.7929	23.0479	24.7174	25.0069	21.4798	21.6010	15.2340	15.2624
	32	21.0784	21.2475	24.3869	24.6355	21.8945	22.0825	12.1560	12.1749
	16	22.1018	22.3923	22.4417	22.7423	22.5449	22.9411	8.5876	8.5981
cameraman	64	22.7777	22.9708	22.1233	22.3349	21.3581	21.5975	17.2186	17.3436
	32	23.6710	23.8155	22.4618	22.6705	20.2102	20.3751	16.5566	16.6426
	16	22.5082	22.7939	22.1487	22.4109	20.0316	20.2916	10.8002	10.8098
Zelda	64	15.7277	15.7677	15.7782	15.8170	16.0613	16.1144	15.5395	15.6069
	32	15.7000	15.7433	15.4379	15.4819	15.7146	15.7761	17.9931	18.1382

	16	16.4336	16.5067	16.0100	16.0869	15.0869	15.1864	16.2196	16.3247
Goldhill	64	24.5843	24.6350	24.2822	24.3547	24.0841	24.2317	17.1520	17.1720
	32	25.0871	25.0962	24.6382	24.7779	24.9298	25.1073	20.4776	20.6218
	16	24.8331	25.0245	24.6934	24.9229	24.0491	24.5184	12.3862	12.4118
Mean	64	21.6708	21.8344	21.8593	22.0187	21.0606	21.2302	17.4947	17.5815
	32	21.7839	21.8228	21.2602	21.3753	20.4248	20.6093	17.7015	17.8397
	16	21.6677	21.9010	22.1141	22.4028	20.5137	20.8385	11.6889	11.7225

Table 4.14: The values of RMSE of each individual image with image block size $B \in \{64, 32, 16\}$ recovered via SVR CPA sensing matrices from $SR \in \{0.25, 0.1, 0.04, 0.01\}$, with and without applying BM3D. The lowest value in each setting is in bold.

Images	B	RMSE							
		SR = 0.25		0.1		0.04		0.01	
		w/o BM3D	w/ BM3D	w/o BM3D	w/ BM3D	w/o BM3D	w/ BM3D	w/o BM3D	w/ BM3D
baboon	64	0.1217	0.1208	0.0933	0.0936	0.1232	0.1223	0.1236	0.1226
	32	0.1051	0.1099	0.0970	0.0988	0.1155	0.1144	0.1248	0.1230
	16	0.1136	0.1128	0.1302	0.1298	0.1073	0.1067	0.2050	0.2032
Barbara	64	0.0764	0.0758	0.0749	0.0738	0.0900	0.0889	0.1358	0.1347
	32	0.1640	0.1653	0.1420	0.1423	0.0874	0.0847	0.1244	0.1222
	16	0.0779	0.0763	0.0796	0.0772	0.0869	0.0827	0.2223	0.2209
cat	64	0.1111	0.1104	0.1143	0.1091	0.0858	0.0844	0.1064	0.1050
	32	0.0581	0.0596	0.1021	0.1025	0.0890	0.0880	0.1088	0.1067
	16	0.1057	0.1047	0.0717	0.0706	0.0798	0.0772	0.2460	0.2454
fruits	64	0.0660	0.0636	0.0703	0.0681	0.0911	0.0890	0.1733	0.1723
	32	0.0571	0.0565	0.0705	0.0688	0.2111	0.2115	0.1724	0.1706
	16	0.0697	0.0670	0.0723	0.0693	0.1706	0.1697	0.3161	0.3155
Lenna	64	0.0624	0.0600	0.0591	0.0567	0.0664	0.0641	0.0986	0.0967
	32	0.0562	0.0534	0.0666	0.0646	0.0707	0.0679	0.0999	0.0972
	16	0.0649	0.0616	0.0487	0.0449	0.0672	0.0622	0.3024	0.3020
peppers	64	0.0674	0.0656	0.0816	0.0800	0.0721	0.0697	0.1061	0.1041

	32	0.0755	0.0740	0.0788	0.0766	0.0628	0.0595	0.1109	0.1082
	16	0.0682	0.0653	0.0616	0.0581	0.0791	0.0744	0.3353	0.3350
airplane	64	0.0725	0.0704	0.0581	0.0562	0.0843	0.0832	0.1731	0.1725
	32	0.0883	0.0866	0.0603	0.0586	0.0804	0.0787	0.2467	0.2462
	16	0.0785	0.0759	0.0755	0.0729	0.0746	0.0713	0.3721	0.3716
cameraman	64	0.0726	0.0710	0.0783	0.0764	0.0855	0.0832	0.1377	0.1358
	32	0.0655	0.0645	0.0753	0.0735	0.0976	0.0958	0.1487	0.1472
	16	0.0749	0.0725	0.0781	0.0758	0.0996	0.0967	0.2884	0.2881
Zelda	64	0.1635	0.1628	0.1626	0.1619	0.1574	0.1564	0.1671	0.1658
	32	0.1641	0.1632	0.1691	0.1682	0.1638	0.1626	0.1260	0.1239
	16	0.1508	0.1495	0.1583	0.1569	0.1761	0.1741	0.1545	0.1527
Goldhill	64	0.0590	0.0586	0.0611	0.0606	0.0625	0.0614	0.1388	0.1385
	32	0.0557	0.0556	0.0586	0.0577	0.0567	0.0555	0.0946	0.0931
	16	0.0573	0.0561	0.0583	0.0567	0.0627	0.0594	0.2403	0.2396
Mean	64	0.0873	0.0859	0.0854	0.0836	0.0918	0.0903	0.1360	0.1348
	32	0.0890	0.0889	0.0920	0.0912	0.1035	0.1019	0.1357	0.1338
	16	0.0861	0.0842	0.0834	0.0812	0.1004	0.0974	0.2682	0.2674

Table 4.15: The values of SSIM of each individual image with image block size $B \in \{64, 32, 16\}$ recovered via SVR CPA sensing matrices from $SR \in \{0.25, 0.1, 0.04, 0.01\}$, with and without applying BM3D. The highest value in each setting is in bold.

Images	B	SSIM							
		SR = 0.25		0.1		0.04		0.01	
		w/o BM3D	w/ BM3D	w/o BM3D	w/ BM3D	w/o BM3D	w/ BM3D	w/o BM3D	w/ BM3D
baboon	64	0.5231	0.4737	0.4693	0.4225	0.3696	0.3549	0.2534	0.2537
	32	0.5405	0.4272	0.4526	0.3874	0.3585	0.3403	0.2500	0.2558
	16	0.5157	0.4670	0.4605	0.4223	0.3470	0.3286	0.1931	0.1977
Barbara	64	0.6460	0.6342	0.6292	0.6287	0.5703	0.5871	0.4261	0.4457
	32	0.6363	0.5931	0.6052	0.5911	0.5584	0.5850	0.3985	0.4268
	16	0.6242	0.6301	0.5955	0.6170	0.5107	0.5536	0.2851	0.3034
cat	64	0.6996	0.6811	0.6691	0.6625	0.5983	0.6123	0.4636	0.4831
	32	0.7340	0.6918	0.6577	0.6371	0.5860	0.5954	0.4639	0.4899

	16	0.6951	0.6835	0.6706	0.6656	0.5832	0.6004	0.3766	0.3972
fruits	64	0.7557	0.7426	0.7224	0.7168	0.6019	0.6247	0.3888	0.4122
	32	0.7981	0.7734	0.7150	0.7142	0.5195	0.5259	0.3649	0.3942
	16	0.7365	0.7401	0.7062	0.7179	0.5245	0.5481	0.2521	0.2669
Lenna	64	0.7974	0.7978	0.7814	0.7881	0.7199	0.7468	0.5454	0.5764
	32	0.8157	0.8166	0.7883	0.7955	0.7067	0.7377	0.5435	0.5829
	16	0.7653	0.7876	0.7665	0.7905	0.6623	0.7106	0.3630	0.3787
peppers	64	0.7936	0.7950	0.7758	0.7834	0.7108	0.7419	0.5214	0.5538
	32	0.8022	0.8046	0.7683	0.7860	0.6990	0.7335	0.5120	0.5503
	16	0.7609	0.7880	0.7431	0.7788	0.6368	0.6924	0.3240	0.3397
airplane	64	0.8158	0.8138	0.7915	0.7953	0.7095	0.7305	0.5636	0.5849
	32	0.8299	0.8282	0.7895	0.7936	0.7009	0.7254	0.5385	0.5624
	16	0.7942	0.8089	0.7741	0.7940	0.6929	0.7236	0.4161	0.4408
cameraman	64	0.7398	0.7295	0.7225	0.7216	0.6393	0.6659	0.4395	0.4696
	32	0.7804	0.7559	0.7280	0.7230	0.6500	0.6615	0.4954	0.5178
	16	0.7347	0.7269	0.7262	0.7227	0.6242	0.6389	0.3911	0.4080
Zelda	64	0.7976	0.8086	0.7903	0.8028	0.7349	0.7654	0.5733	0.6087
	32	0.8045	0.8191	0.7687	0.7924	0.7120	0.7541	0.5717	0.6179
	16	0.7567	0.8056	0.7276	0.7875	0.6187	0.7025	0.4308	0.4634
Goldhill	64	0.6960	0.6704	0.6711	0.6540	0.5908	0.5955	0.4498	0.4577
	32	0.7298	0.6931	0.6730	0.6582	0.5849	0.5891	0.4425	0.4614
	16	0.6845	0.6721	0.6572	0.6557	0.5618	0.5846	0.3301	0.3465
Mean	64	0.7265	0.7147	0.7023	0.6976	0.6245	0.6425	0.4625	0.4846
	32	0.7471	0.7203	0.6946	0.6879	0.6076	0.6248	0.4581	0.4859
	16	0.7068	0.7110	0.6828	0.6952	0.5762	0.6083	0.3362	0.3542

Table 4.16: The time cost for image recovery (in seconds) of each individual image with image block size $B \in \{64, 32, 16\}$ recovered via SVR CPA sensing matrices from $SR \in \{0.25, 0.1, 0.04, 0.01\}$, without applying BM3D. The lowest time cost in each setting is in bold.

Images	B	Time cost for image recovery (s)			
		SR = 0.25	0.1	0.04	0.01

baboon	64	59.77	37.81	28.23	17.75
	32	24.42	17.56	16.30	18.00
	16	38.75	37.22	38.66	65.91
Barbara	64	70.11	38.14	31.94	19.45
	32	24.77	19.47	18.06	19.02
	16	36.52	38.84	41.89	58.05
cat	64	55.27	34.22	28.63	20.06
	32	22.27	15.75	16.09	20.13
	16	36.84	37.25	44.17	64.16
fruits	64	68.63	42.38	33.92	19.42
	32	26.17	19.50	19.00	20.14
	16	43.69	44.08	44.94	58.70
Lenna	64	62.00	36.36	29.63	18.02
	32	25.11	16.73	16.48	18.38
	16	35.98	38.34	39.08	64.31
peppers	64	62.91	38.06	30.45	19.31
	32	22.02	16.38	16.69	17.92
	16	36.55	37.36	40.30	63.38
airplane	64	51.50	31.00	27.14	17.58
	32	19.69	14.53	15.28	17.14
	16	32.58	33.67	37.47	58.53
cameraman	64	14.16	9.70	8.17	5.05
	32	5.64	4.25	4.28	4.78
	16	9.08	9.39	9.94	15.00
Zelda	64	56.61	35.44	28.19	18.00
	32	22.77	16.59	16.70	18.72
	16	37.45	37.56	39.66	58.92
Goldhill	64	57.98	36.05	28.89	17.66
	32	22.66	16.39	16.08	17.41
	16	37.39	36.72	38.48	58.22
Mean	64	55.89	33.92	27.52	17.23
	32	21.55	15.72	15.50	17.16
	16	34.48	35.04	37.46	56.52

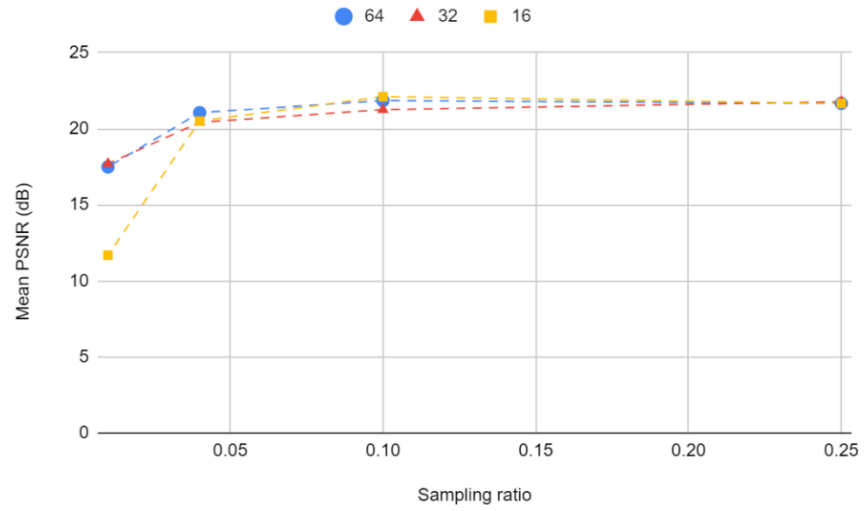


Figure 4.19: The values of mean PSNR of the images with image block size $B \in \{64, 32, 16\}$ recovered via SVR CPA sensing matrices.

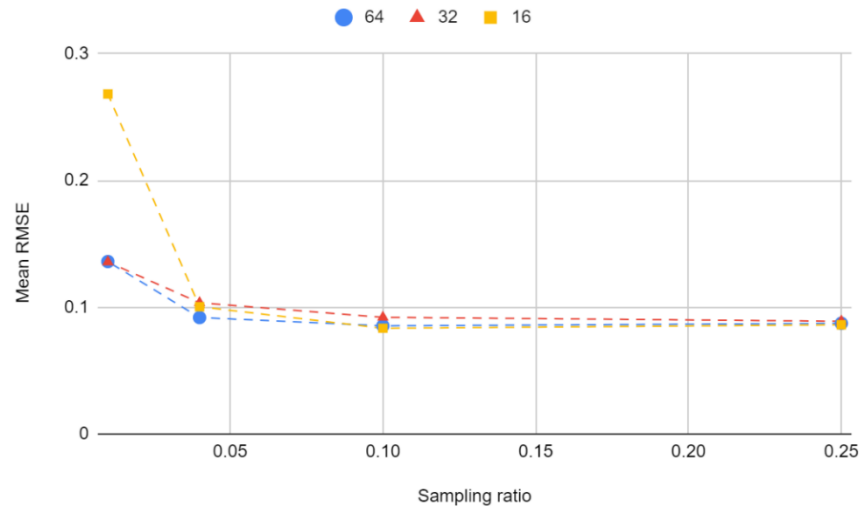


Figure 4.20: The values of mean RMSE of the images with image block size $B \in \{64, 32, 16\}$ recovered via SVR CPA sensing matrices.

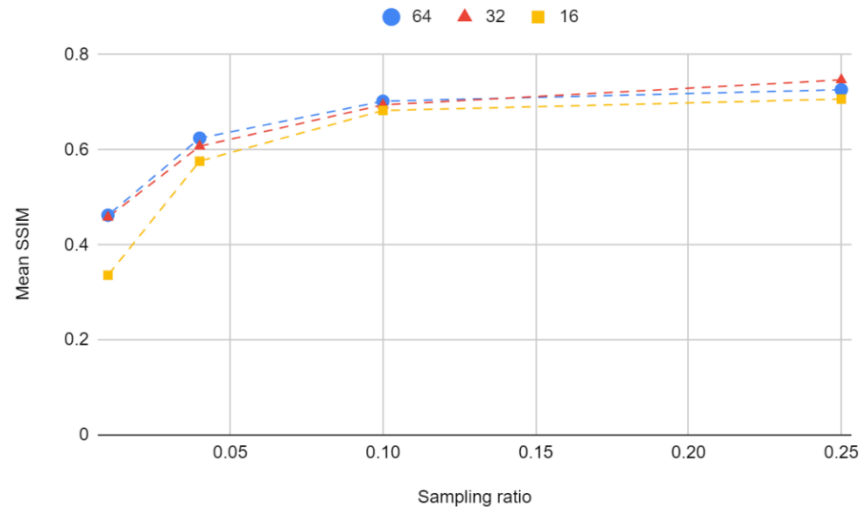


Figure 4.21: The values of mean SSIM of the images with image block size $B \in \{64, 32, 16\}$ recovered via SVR CPA sensing matrices.

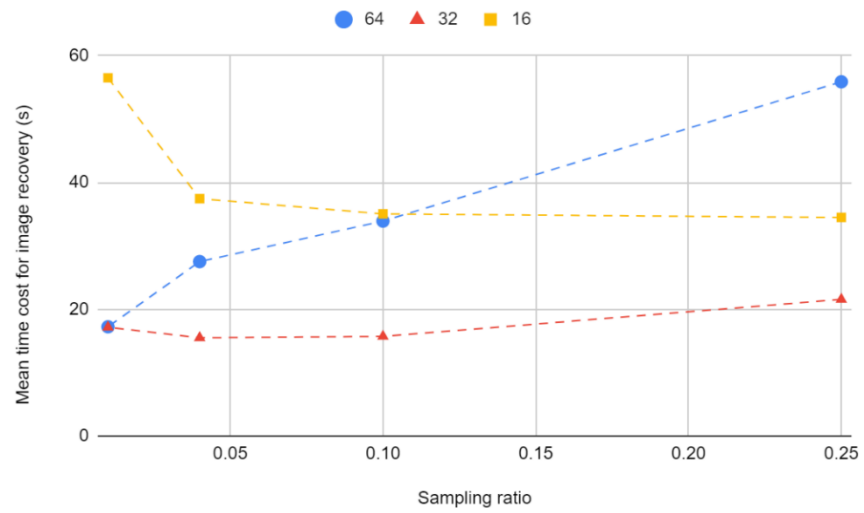


Figure 4.22: The mean time cost for image recovery of images with image block size $B \in \{64, 32, 16\}$ recovered via SVR CPA sensing matrices.

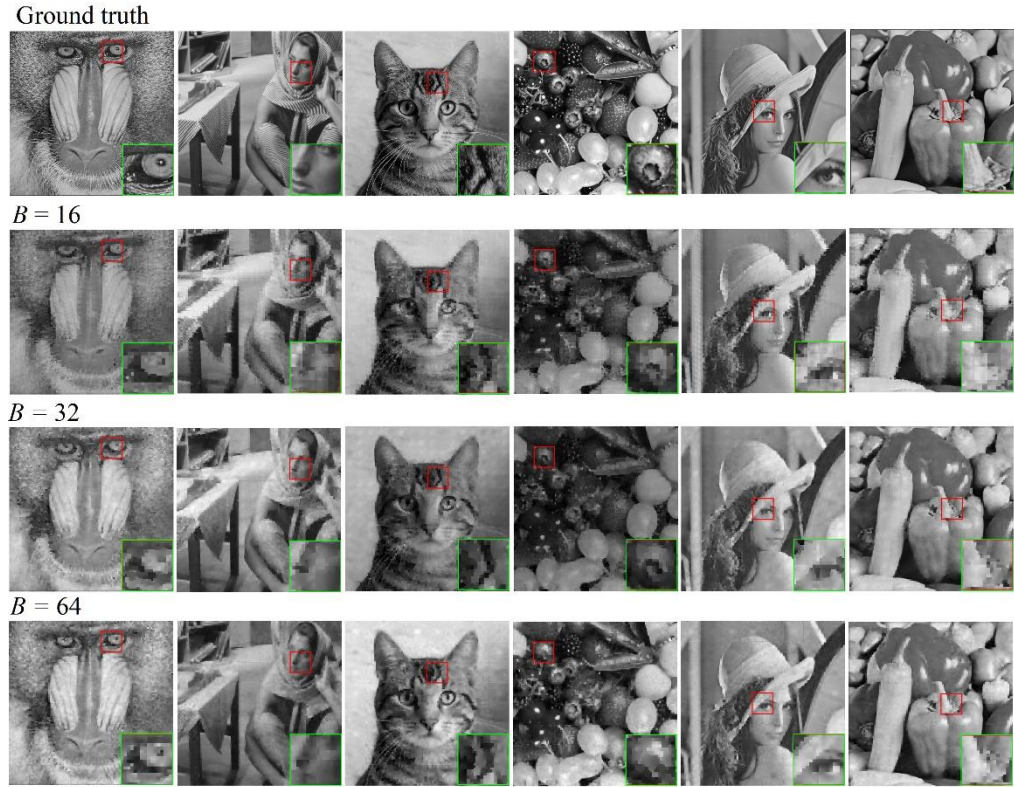


Figure 4.23: (First row) The ground truth of our testing images. From left to right, "baboon", "Barbara", "cat", "fruits", "Lena", and "peppers". All images are of size 512×512 . (Second row) The images formed from the image blocks of size 16×16 recovered via SVR CPA sensing matrices from $SR = 0.1$. (Third row) The images formed from the image blocks of size 32×32 recovered via SVR CPA sensing matrices from $SR = 0.1$. (Fourth row) The images formed from the image blocks of size 64×64 recovered via SVR CPA sensing matrices from $SR = 0.1$.

The results in Table 4.13 – 4.16 and Figure 4.19 – 4.22 show that the images with image blocks size $B \in \{64, 32, 16\}$ recovered via SVR CPA sensing matrices have comparable values of mean PSNR, RMSE, and SSIM except for $SR = 0.01$, where the image quality of images with image block size $B = 16$ is lower than that of images with image block size $B \in \{64, 32\}$. The

mean time cost for image recovery of images with image block size $B = 32$ is significantly lower than that of images with image block size $B \in \{64,16\}$. The results indicate that $B = 32$ is the optimum image block size since for a comparable image quality it gives the lowest time cost for image blocks recovery compared to the other image block size tested in the numerical experiments. All the following simulations use block size of 32×32 .

4.2.2 Comparison of SVR RBP, HMP, And CPA Sensing Matrices for Block-Based CS

This section compares the performance of different types of SVR sensing matrices extensively. Same as in Section 4.1.2, the SVR sensing matrices of choice are RBP, HMP, and CPA. Table 4.17 – 4.20 show the values of PSNR (in dB), RMSE, SSIM, and time cost for image recovery (in seconds) of the images recovered via SVR RBP, HMP, and CPA sensing matrices from $SR \in \{0.25, 0.1, 0.04, 0.01\}$, with and without applying BM3D, for block-based approach. Figure 4.24 – 4.27 show the values of mean PSNR (in dB), RMSE, SSIM, and time cost for image recovery (in seconds) of the images recovered via SVR RBP, HMP, and CPA sensing matrices from $SR \in \{0.25, 0.1, 0.04, 0.01\}$, without applying BM3D, for block-based approach.

Table 4.17: The values of PSNR (in dB) of each individual image recovered via SVR RBP, HMP, and CPA sensing matrices from $SR \in \{0.25, 0.1, 0.04, 0.01\}$, with and without applying BM3D, for block-based approach. The highest value in each setting is in bold.

Images	Sensing matrices	PSNR (dB)							
		SR = 0.25		0.1		0.04		0.01	
		w/o BM3D	w/ BM3D	w/o BM3D	w/ BM3D	w/o BM3D	w/ BM3D	w/o BM3D	w/ BM3D
baboon	RBP	19.0872	19.0024	15.8973	15.9099	18.2712	18.3607	18.2632	18.3518
	HMP	19.0847	18.9629	20.6255	20.4590	18.6228	18.6482	15.5140	15.5646
	CPA	20.4220	19.9904	20.2632	20.1072	18.7481	18.8346	18.0764	18.2051
Barbara	RBP	21.6125	21.7863	21.5531	21.7733	21.3936	21.6207	18.3192	18.4768
	HMP	22.4482	22.5339	21.6956	21.8276	20.8773	21.0961	18.3581	18.4807
	CPA	17.5522	17.4628	16.9564	16.9383	21.1707	21.4387	18.1059	18.2600
cat	RBP	20.1823	20.2446	18.6508	18.7359	19.8964	20.0473	18.2296	18.3614
	HMP	19.7860	19.8223	20.6790	20.7542	21.6741	21.7894	17.9339	18.0236
	CPA	24.8930	24.6888	19.8230	19.7902	21.0106	21.1109	19.2643	19.4372
fruits	RBP	20.3792	20.3716	21.5871	21.7613	19.6798	19.8567	16.3398	16.4764
	HMP	22.9440	23.0033	21.5288	21.6052	16.8108	16.8243	14.5016	14.5304
	CPA	24.6767	24.7337	23.0320	23.2473	13.5095	13.4945	15.2705	15.3579
Lena	RBP	24.1152	24.5284	24.0763	24.5035	22.3963	22.7099	15.1139	15.1520
	HMP	24.0158	24.3612	25.1573	25.4833	22.4075	22.6140	19.6083	19.7314
	CPA	25.9046	26.3758	23.5326	23.7905	23.0157	23.3656	20.0064	20.2385
peppers	RBP	23.0686	23.3160	22.4083	22.4083	23.0105	23.3938	19.1625	19.3792
	HMP	23.7145	24.0167	24.1683	24.4752	22.1089	22.4375	18.5627	18.6917
	CPA	22.6749	22.8633	22.0697	22.3122	24.0429	24.5062	19.1021	19.3151
airplane	RBP	24.2650	24.3708	21.8861	22.0300	20.0518	20.1551	13.3376	13.3658
	HMP	24.6723	24.9537	24.1583	24.2713	21.9590	22.1399	16.7606	16.8068
	CPA	21.0784	21.2475	24.3869	24.6355	21.8945	22.0825	12.3712	12.3916
cameraman	RBP	22.4970	22.6400	21.6421	21.7930	20.2299	20.3877	17.2281	17.3372
	HMP	21.9928	22.0350	21.1531	21.3195	19.4120	19.5485	16.8380	16.9003
	CPA	23.6710	23.8155	22.4618	22.6705	20.2102	20.3751	16.5566	16.6425
Zelda	RBP	15.9718	16.0262	15.8320	15.8914	15.0838	15.1505	20.3510	20.5610
	HMP	17.2224	17.2857	17.4898	17.5534	16.2020	16.2807	13.8630	13.9131

	CPA	15.7000	15.7433	15.4379	15.4819	15.7146	15.7761	17.9938	18.1390
Goldhill	RBP	23.9280	24.0374	23.6677	23.8317	23.2590	23.5463	20.9866	21.1793
	HMP	24.5715	24.6218	23.8798	23.9234	23.2285	23.3981	20.6787	20.8052
	CPA	25.0871	25.0962	24.6382	24.7779	24.9298	25.1070	20.4780	20.6224
Mean	RBP	21.5107	21.6324	20.7201	20.8638	20.3272	20.5229	17.7331	17.8641
	HMP	22.0452	22.1597	22.0535	22.1672	20.3303	20.4777	17.2619	17.3448
	CPA	22.1660	22.2017	21.2602	21.3752	20.4247	20.6091	17.7225	17.8609

Table 4.18: The values of RMSE of each individual image recovered via SVR RBP, HMP, and CPA sensing matrices from $SR \in \{0.25, 0.1, 0.04, 0.01\}$, with and without applying BM3D, for block-based approach. The lowest value in each setting is in bold.

Images	Sensing matrices	RMSE							
		SR = 0.25		0.1		0.04		0.01	
		w/o BM3D	w/ BM3D	w/o BM3D	w/ BM3D	w/o BM3D	w/ BM3D	w/o BM3D	w/ BM3D
baboon	RBP	0.1111	0.1122	0.1604	0.1601	0.1220	0.1208	0.1221	0.1209
	HMP	0.1111	0.1127	0.0931	0.0949	0.1172	0.1172	0.1676	0.1666
	CPA	0.0953	0.1001	0.0970	0.0988	0.1155	0.1144	0.1248	0.1230
Barbara	RBP	0.0831	0.0814	0.0836	0.0815	0.0852	0.0830	0.1213	0.1192
	HMP	0.0754	0.0747	0.0823	0.0810	0.0904	0.0881	0.1208	0.1191
	CPA	0.1326	0.1339	0.1420	0.1423	0.0874	0.0847	0.1244	0.1222
cat	RBP	0.0979	0.0972	0.1168	0.1157	0.1012	0.0995	0.1226	0.1208
	HMP	0.1025	0.1021	0.0925	0.0917	0.0825	0.0814	0.1269	0.1256
	CPA	0.0569	0.0583	0.1021	0.1024	0.0890	0.0880	0.1088	0.1067
fruits	RBP	0.0957	0.0958	0.0833	0.0816	0.1038	0.1017	0.1524	0.1500
	HMP	0.0713	0.0708	0.0839	0.0831	0.1444	0.1441	0.1883	0.1877
	CPA	0.0584	0.0580	0.0705	0.0688	0.2111	0.2115	0.1724	0.1706
Lena	RBP	0.0623	0.0594	0.0625	0.0595	0.0759	0.0732	0.1755	0.1747
	HMP	0.0630	0.0605	0.0552	0.0532	0.0758	0.0740	0.1046	0.1031
	CPA	0.0507	0.0480	0.0666	0.0646	0.0707	0.0679	0.0999	0.0973
peppers	RBP	0.0702	0.0683	0.0758	0.0734	0.0707	0.0677	0.1101	0.1074
	HMP	0.0652	0.0630	0.0619	0.0597	0.0784	0.0755	0.1180	0.1163
	CPA	0.0735	0.0719	0.0788	0.0766	0.0628	0.0595	0.1109	0.1082

airplane	RBP	0.0612	0.0605	0.0805	0.0792	0.0994	0.0982	0.2153	0.2146
	HMP	0.0584	0.0565	0.0620	0.0612	0.0798	0.0782	0.1452	0.1444
	CPA	0.0883	0.0866	0.0603	0.0586	0.0804	0.0787	0.2407	0.2401
cameraman	RBP	0.0750	0.0738	0.0828	0.0813	0.0974	0.0956	0.1376	0.1359
	HMP	0.0795	0.0791	0.0876	0.0859	0.1070	0.1053	0.1439	0.1429
	CPA	0.0655	0.0644	0.0753	0.0735	0.0976	0.0958	0.1487	0.1472
Zelda	RBP	0.1590	0.1580	0.1616	0.1605	0.1761	0.1748	0.0960	0.0937
	HMP	0.1377	0.1367	0.1335	0.1325	0.1548	0.1534	0.2027	0.2015
	CPA	0.1641	0.1632	0.1691	0.1682	0.1638	0.1626	0.1260	0.1239
Goldhill	RBP	0.0636	0.0628	0.0656	0.0643	0.0687	0.0665	0.0893	0.0873
	HMP	0.0591	0.0587	0.0640	0.0637	0.0690	0.0676	0.0925	0.0911
	CPA	0.0557	0.0556	0.0586	0.0577	0.0567	0.0555	0.0946	0.0931
Mean	RBP	0.0879	0.0869	0.0973	0.0957	0.1000	0.0981	0.1342	0.1325
	HMP	0.0823	0.0815	0.0816	0.0807	0.0999	0.0985	0.1411	0.1398
	CPA	0.0841	0.0840	0.0920	0.0912	0.1035	0.1019	0.1351	0.1332

Table 4.19: The values of SSIM of each individual image recovered via SVR RBP, HMP, and CPA sensing matrices from $SR \in \{0.25, 0.1, 0.04, 0.01\}$, with and without applying BM3D, for block-based approach. The highest value in each setting is in bold.

Images	Sensing matrices	SSIM							
		SR = 0.25		0.1		0.04		0.01	
		w/o BM3D	w/ BM3D	w/o BM3D	w/ BM3D	w/o BM3D	w/ BM3D	w/o BM3D	w/ BM3D
baboon	RBP	0.4150	0.3735	0.3692	0.3455	0.3348	0.3251	0.2520	0.2530
	HMP	0.5149	0.4396	0.4426	0.3834	0.3400	0.3189	0.2463	0.2442
	CPA	0.5392	0.4302	0.4525	0.3873	0.3584	0.3402	0.2500	0.2557
Barbara	RBP	0.5952	0.6066	0.5733	0.5926	0.5373	0.5633	0.3941	0.4168
	HMP	0.6356	0.6273	0.5867	0.5935	0.4985	0.5303	0.3877	0.4148
	CPA	0.6657	0.6819	0.6395	0.6572	0.5643	0.5951	0.4075	0.4318
cat	RBP	0.6568	0.6486	0.6155	0.6190	0.5784	0.5913	0.4817	0.4817
	HMP	0.6905	0.6709	0.6433	0.6383	0.5685	0.5799	0.4832	0.4989
	CPA	0.7344	0.6931	0.6579	0.6374	0.5862	0.5956	0.4641	0.4901
fruits	RBP	0.6882	0.6804	0.6353	0.6477	0.5458	0.5756	0.3669	0.3908

	HMP	0.7381	0.7240	0.6639	0.6647	0.5127	0.5349	0.3709	0.3931
	CPA	0.7972	0.7713	0.7149	0.7141	0.5194	0.5258	0.3648	0.3941
Lena	RBP	0.7566	0.7714	0.7289	0.7486	0.6916	0.7201	0.5451	0.5660
	HMP	0.7650	0.7780	0.7320	0.7495	0.6351	0.6709	0.5382	0.5698
	CPA	0.8228	0.8208	0.7878	0.7878	0.7062	0.7372	0.5430	0.5823
peppers	RBP	0.7496	0.7679	0.7186	0.7439	0.6734	0.7084	0.5151	0.5454
	HMP	0.7642	0.7807	0.7241	0.7493	0.6134	0.6612	0.4966	0.5346
	CPA	0.8038	0.8046	0.7679	0.7854	0.6986	0.7330	0.5116	0.5497
airplane	RBP	0.7688	0.7698	0.7351	0.7450	0.6882	0.7038	0.5743	0.5924
	HMP	0.7911	0.7918	0.7504	0.7583	0.6582	0.6843	0.5795	0.6015
	CPA	0.8299	0.8282	0.7895	0.7936	0.7009	0.7254	0.5399	0.5642
cameraman	RBP	0.7190	0.7159	0.6914	0.6931	0.6323	0.6445	0.5141	0.5325
	HMP	0.7416	0.7254	0.6933	0.6943	0.5959	0.6148	0.4969	0.5155
	CPA	0.7804	0.7559	0.7280	0.7230	0.6500	0.6615	0.4953	0.5177
Zelda	RBP	0.7464	0.7816	0.7189	0.7636	0.6575	0.7112	0.5874	0.6310
	HMP	0.7835	0.8085	0.7557	0.7878	0.6530	0.7100	0.5233	0.5712
	CPA	0.8045	0.8191	0.7687	0.7924	0.7120	0.7541	0.5717	0.6179
Goldhill	RBP	0.6408	0.6345	0.6049	0.6084	0.5512	0.5694	0.4450	0.4619
	HMP	0.6877	0.6574	0.6406	0.6217	0.5489	0.5564	0.4408	0.4569
	CPA	0.7298	0.6931	0.6730	0.6582	0.5849	0.5891	0.4425	0.4613
Mean	RBP	0.6737	0.6750	0.6391	0.6507	0.5890	0.6113	0.4676	0.4872
	HMP	0.7112	0.7004	0.6633	0.6641	0.5624	0.5862	0.4563	0.4800
	CPA	0.7508	0.7298	0.6980	0.6936	0.6081	0.6257	0.4590	0.4865

Table 4.20: The time cost for image recovery of each individual image recovered via RBP, HMP, and CPA SVR sensing matrices from $SR \in \{0.25, 0.1, 0.04, 0.01\}$, without applying BM3D, for block-based approach. The lowest time cost in each setting are in bold.

Images	Sensing matrices	Time cost for recovery (s)			
		SR = 0.25	0.1	0.04	0.01
baboon	RBP	23.33	16.39	20.16	14.05
	HMP	17.28	15.72	18.77	13.86
	CPA	27.27	23.14	16.72	24.23

Barbara	RBP	30.27	22.36	17.39	14.53
	HMP	44.91	22.36	29.14	31.80
	CPA	29.36	18.17	22.95	19.19
cat	RBP	34.72	25.95	15.33	13.73
	HMP	27.53	27.42	24.80	20.58
	CPA	26.47	18.36	24.05	19.09
fruits	RBP	37.92	24.13	25.97	18.17
	HMP	40.30	32.81	29.31	26.33
	CPA	32.61	25.02	27.28	23.11
Lena	RBP	32.77	27.38	19.86	15.20
	HMP	29.86	21.33	19.89	28.86
	CPA	24.44	17.16	20.94	24.98
peppers	RBP	36.09	18.98	16.16	15.09
	HMP	33.38	30.25	22.95	33.63
	CPA	24.22	17.38	18.59	19.22
airplane	RBP	22.34	13.47	12.73	12.55
	HMP	16.20	12.97	11.42	8.88
	CPA	21.45	15.28	16.34	18.41
cameraman	RBP	6.13	4.58	4.20	4.03
	HMP	6.17	4.08	3.72	3.02
	CPA	6.17	5.02	4.94	5.05
Zelda	RBP	25.28	18.30	16.92	14.95
	HMP	21.23	14.83	13.55	12.25
	CPA	24.92	17.97	17.86	20.09
Goldhill	RBP	28.30	16.59	14.58	12.78
	HMP	18.16	13.33	11.28	9.38
	CPA	26.53	17.66	16.67	18.61
Mean	RBP	27.71	18.81	16.33	13.51
	HMP	25.50	19.51	18.48	18.86
	CPA	24.34	17.51	18.63	19.20

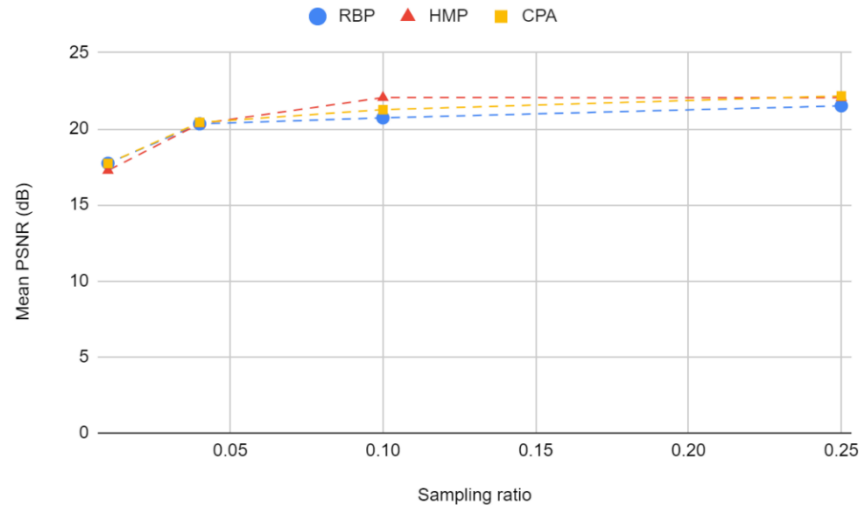


Figure 4.24: Mean values of PSNR (in dB) of image recovered via SVR RBP, HMP, and CPA sensing matrices from $SR \in \{0.25, 0.1, 0.04, 0.01\}$, without applying BM3D, for block-based approach.

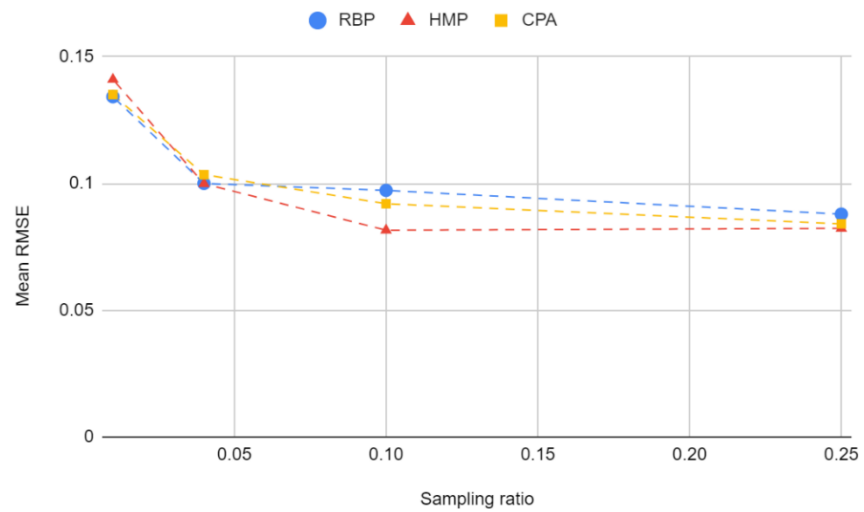


Figure 4.25: Mean values of RMSE of image recovered via SVR RBP, HMP, and CPA sensing matrices from $SR \in \{0.25, 0.1, 0.04, 0.01\}$, without applying BM3D, for block-based approach.

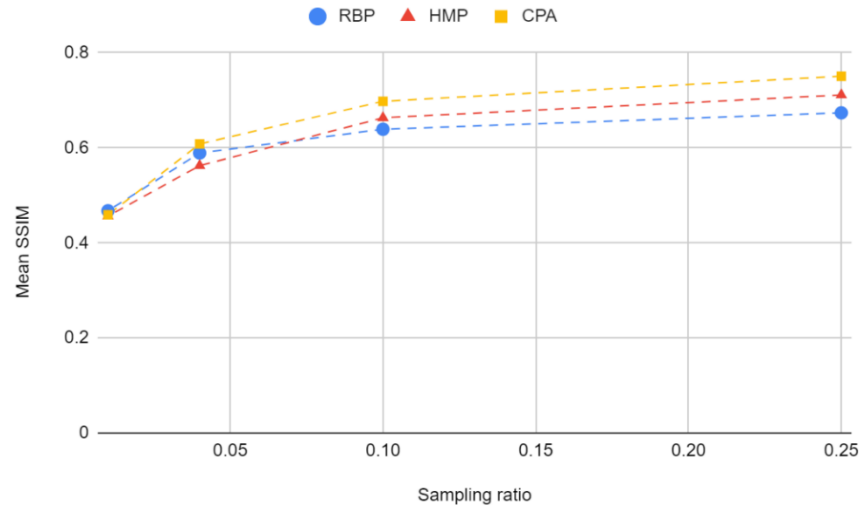


Figure 4.26: Mean values of SSIM of image recovered via SVR RBP, HMP, and CPA sensing matrices from $SR \in \{0.25, 0.1, 0.04, 0.01\}$, without applying BM3D, for block-based approach.

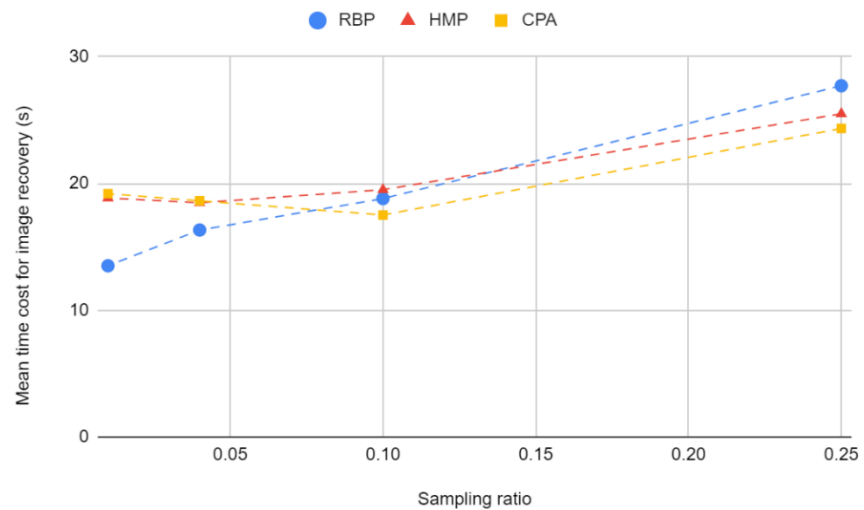


Figure 4.27: Mean time cost for image recovery of image recovered via SVR RBP, HMP, and CPA sensing matrices from $SR \in \{0.25, 0.1, 0.04, 0.01\}$, without applying BM3D, for block-based approach.

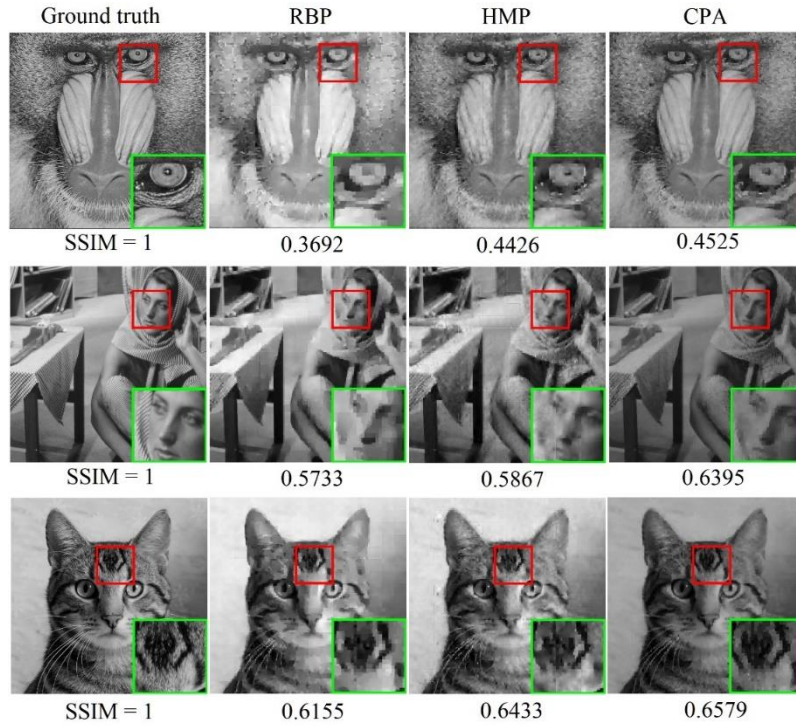


Figure 4.28: (Start from left, first column) The original images of “baboon”, “Barbara”, and “cat”. (Second to fourth columns) “baboon”, “Barbara”, and “cat” recovered via SVR RBP, HMP, CPA sensing matrices from $SR = 0.1$, without applying BM3D, for block-based approach with block size 32×32 . All images are shown with their respective values of SSIM.

Based on the results in Table 4.17 – 4.20 and Figure 4.24 – 4.27, it can be seen that overall, the values of mean PSNR (in dB) of images recovered via SVR RBP, HMP, and CPA sensing matrices from all values of SR tested without applying BM3D are comparable to each other. The values of mean RMSE for images recovered via SVR RBP, HMP, and CPA sensing matrices from all values of SR are comparable except for $SR = 0.1$, where the value of mean RMSE for SVR HMP is the lowest. The value of mean SSIM is the highest for images recovered via SVR CPA sensing matrices from all values of SR

tested. The mean time cost for image recovery (in seconds) is the lowest for images recovered from SVR RBP sensing matrices from $SR = 0.04$ and 0.01 , and it is the lowest for images recovered from SVR CPA sensing matrices from $SR = 0.25$ and 0.1 .

In addition, Figure 4.28 shows that the image details of images recovered via SVR CPA sensing matrices are clearer than that of images recovered via SVR RBP and HMP sensing matrices for the same value of SR . Hence, SVR CPA sensing matrices are able to improve the image quality better than SVR RBP and HMP sensing matrices. The results shown indicate that CPA sensing matrices is the more suitable choice for SVR block-based approach.

4.2.3 Comparison of UR and SVR CPA Sensing Matrices for Block-Based CS

In this section, the image quality of the images recovered via UR and SVR CPA sensing matrices for block-based approach was examined extensively. After the initial images are recovered, BM3D was applied as the image denoising algorithm to remove the edges of the image blocks and noises in the recovered images. Table 4.21 – 4.24 show the values of PSNR (in dB), RMSE, SSIM, and time cost for image recovery (in seconds) of images recovered via UR and SVR CPA sensing matrices, with and without applying BM3D. Figure 4.29 – 4.32 show the values of mean PSNR (in dB), RMSE, SSIM, and time cost for image recovery (in seconds) of images recovered via UR and SVR CPA sensing matrices, without applying BM3D. Furthermore, Figure 4.33 shows images recovered via UR and SVR CPA sensing matrices from $SR = 0.1$. To demonstrate the effects to the images after applying BM3D, Figure 4.34 shows

"peppers" recovered via SVR CPA sensing matrices from SR = 0.1, with and without applying BM3D.

Table 4.21: The values of PSNR (in dB) of each individual image recovered via UR and SVR CPA sensing matrices, with and without applying BM3D.

The highest values in each setting is in bold.

Images	Resolution	PSNR (dB)							
		SR = 0.25		0.1		0.04		0.01	
		w/o BM3D	w/ BM3D	w/o BM3D	w/ BM3D	w/o BM3D	w/ BM3D	w/o BM3D	w/ BM3D
baboon	UR	19.2277	18.9105	15.0720	15.0384	10.3273	10.3202	16.4542	16.5416
	SVR	20.4220	19.9904	20.2632	20.1072	18.7481	18.8346	18.0764	18.2051
Barbara	UR	14.7951	14.7306	12.3561	12.3480	13.7316	13.7610	13.3730	13.3967
	SVR	17.5522	17.4628	16.9564	16.9383	21.1707	21.4387	18.1059	18.2600
cat	UR	21.1374	20.9926	18.5780	18.6153	17.7565	17.8509	14.0824	14.1041
	SVR	24.8930	24.6888	19.8230	19.7902	21.0106	21.1109	19.2643	19.4372
fruits	UR	25.6012	25.0283	16.4229	16.3722	11.9250	11.9195	14.2310	14.2904
	SVR	24.6767	24.7337	23.0320	23.2473	13.5095	13.4945	15.2705	15.3579
Lenna	UR	23.7990	23.8931	25.9695	26.4026	17.1269	17.1839	14.8475	14.8742
	SVR	25.9046	26.3758	23.5326	23.7905	23.0157	23.3656	20.0064	20.2385
peppers	UR	23.5520	23.6406	24.4502	24.7912	19.0870	19.2490	14.2613	14.2979
	SVR	22.6749	22.8633	22.0697	22.3122	24.0429	24.5062	19.1021	19.3151
airplane	UR	22.6684	22.5251	18.3832	18.4344	17.7384	17.8389	16.2317	16.2996
	SVR	21.0784	21.2475	24.3869	24.6355	21.8945	22.0825	12.3712	12.3916
cameraman	UR	23.1393	22.9367	17.2761	17.2868	11.1045	11.0969	14.5616	14.6494
	SVR	23.6710	23.8155	22.4618	22.6705	20.2102	20.3751	16.5566	16.6425
Zelda	UR	17.6781	17.7252	16.7689	16.8608	14.5918	14.6870	14.7046	14.7719
	SVR	15.7000	15.7433	15.4379	15.4819	15.7146	15.7761	17.9938	18.1390
Goldhill	UR	28.7532	28.1005	24.8909	25.0467	19.6741	19.7531	18.8188	18.8799
	SVR	25.0871	25.0962	24.6382	24.7779	24.9298	25.1070	20.4780	20.6224
Mean	UR	22.0351	21.8483	19.0168	19.1196	15.3063	15.3660	15.1566	15.2106
	SVR	22.1660	22.2017	21.2602	21.3752	20.4247	20.6091	17.7225	17.8609

Table 4.22: The values of RMSE of each individual image recovered via UR and SVR CPA sensing matrices, with and without applying BM3D. The lowest values in each setting is in bold.

Images	Resolution	RMSE							
		SR = 0.25		0.1		0.04		0.01	
		w/o BM3D	w/ BM3D	w/o BM3D	w/ BM3D	w/o BM3D	w/ BM3D	w/o BM3D	w/ BM3D
baboon	UR	0.1093	0.1134	0.1764	0.1770	0.3045	0.3048	0.1504	0.1489
	SVR	0.0953	0.1001	0.0970	0.0988	0.1155	0.1144	0.1248	0.1230
Barbara	UR	0.1821	0.1834	0.2411	0.2413	0.2058	0.2051	0.2145	0.2139
	SVR	0.1326	0.1339	0.1420	0.1423	0.0874	0.0847	0.1244	0.1222
cat	UR	0.0877	0.0892	0.1178	0.1173	0.1295	0.1281	0.1976	0.1971
	SVR	0.0569	0.0583	0.1021	0.1024	0.0890	0.0880	0.1088	0.1067
fruits	UR	0.0525	0.0561	0.1510	0.1518	0.2534	0.2535	0.1943	0.1930
	SVR	0.0584	0.0580	0.0705	0.0688	0.2111	0.2115	0.1724	0.1706
Lenna	UR	0.0646	0.0639	0.0503	0.0478	0.1392	0.1383	0.1810	0.1804
	SVR	0.0507	0.0480	0.0666	0.0646	0.0707	0.0679	0.0999	0.0973
peppers	UR	0.0664	0.0658	0.0599	0.0576	0.1111	0.1090	0.1936	0.1928
	SVR	0.0735	0.0719	0.0788	0.0766	0.0628	0.0595	0.1109	0.1082
airplane	UR	0.0735	0.0748	0.1205	0.1198	0.1297	0.1282	0.1543	0.1531
	SVR	0.0883	0.0866	0.0603	0.0586	0.0804	0.0787	0.2407	0.2401
cameraman	UR	0.0697	0.0713	0.1368	0.1367	0.2785	0.2787	0.1870	0.1852
	SVR	0.0655	0.0644	0.0753	0.0735	0.0976	0.0958	0.1487	0.1472
Zelda	UR	0.1306	0.1299	0.1451	0.1435	0.1864	0.1844	0.1840	0.1826
	SVR	0.1641	0.1632	0.1691	0.1682	0.1638	0.1626	0.1260	0.1239
Goldhill	UR	0.0365	0.0394	0.0569	0.0559	0.1038	0.1029	0.1146	0.1138
	SVR	0.0557	0.0556	0.0586	0.0577	0.0567	0.0555	0.0946	0.0931
Mean	UR	0.0873	0.0887	0.1256	0.1249	0.1842	0.1833	0.1771	0.1761
	SVR	0.0841	0.0840	0.0920	0.0912	0.1035	0.1019	0.1351	0.1332

Table 4.23: The values of SSIM of each individual image recovered via UR and SVR CPA sensing matrices, with and without applying BM3D. The highest values in each setting is in bold.

Images	Resolution	SSIM							
		SR = 0.25		0.1		0.04		0.01	
		w/o BM3D	w/ BM3D	w/o BM3D	w/ BM3D	w/o BM3D	w/ BM3D	w/o BM3D	w/ BM3D
baboon	UR	0.5751	0.4371	0.3766	0.3202	0.2182	0.2019	0.2286	0.2321
	SVR	0.5392	0.4302	0.4525	0.3873	0.3584	0.3402	0.2500	0.2557
Barbara	UR	0.6344	0.5811	0.4867	0.4779	0.4456	0.4640	0.3541	0.3762
	SVR	0.6610	0.6203	0.6052	0.5911	0.5584	0.5850	0.3984	0.4267
cat	UR	0.7326	0.6818	0.6058	0.6020	0.5149	0.5307	0.4316	0.4460
	SVR	0.7344	0.6931	0.6579	0.6374	0.5862	0.5956	0.4641	0.4901
fruits	UR	0.8136	0.7837	0.6152	0.6154	0.4140	0.4284	0.3288	0.3571
	SVR	0.7972	0.7713	0.7149	0.7141	0.5194	0.5258	0.3648	0.3941
Lenna	UR	0.8436	0.8368	0.7499	0.7717	0.6145	0.6417	0.5101	0.5368
	SVR	0.8228	0.8208	0.7878	0.7878	0.7062	0.7372	0.5430	0.5823
peppers	UR	0.8281	0.8287	0.7340	0.7675	0.5886	0.6324	0.4598	0.4915
	SVR	0.8038	0.8046	0.7679	0.7854	0.6986	0.7330	0.5116	0.5497
airplane	UR	0.8725	0.8473	0.7403	0.7586	0.6055	0.6401	0.4923	0.5232
	SVR	0.8299	0.8282	0.7895	0.7936	0.7009	0.7254	0.5399	0.5642
cameraman	UR	0.8149	0.7796	0.6626	0.6663	0.4764	0.4855	0.4119	0.4403
	SVR	0.7804	0.7559	0.7280	0.7230	0.6500	0.6615	0.4953	0.5177
Zelda	UR	0.8243	0.8393	0.7281	0.7771	0.5763	0.6403	0.4896	0.5308
	SVR	0.8045	0.8191	0.7687	0.7924	0.7120	0.7541	0.5717	0.6179
Goldhill	UR	0.7734	0.7106	0.6316	0.6220	0.5056	0.5159	0.4117	0.4251
	SVR	0.7298	0.6931	0.6730	0.6582	0.5849	0.5891	0.4425	0.4613
Mean	UR	0.7712	0.7326	0.6331	0.6379	0.4960	0.5181	0.4119	0.4359
	SVR	0.7503	0.7237	0.6945	0.6870	0.6075	0.6247	0.4581	0.4860

Table 4.24: The time cost for image recovery (in seconds) of images recovered via UR and SVR CPA sensing matrices, with and without applying BM3D. The lowest time cost in each setting is in bold.

Images	Resolution	Time cost for recovery (s)			
		SR = 0.25	0.1	0.04	0.01
baboon	UR	31.31	23.53	21.45	24.73
	SVR	27.27	23.14	16.72	24.23
Barbara	UR	34.61	25.14	24.13	24.73
	SVR	29.36	18.17	22.95	19.19
cat	UR	28.97	21.20	20.72	35.06
	SVR	26.47	18.36	24.05	19.09
fruits	UR	33.61	22.84	27.98	24.45
	SVR	32.61	25.02	27.28	23.11
Lenna	UR	24.58	23.66	28.66	23.97
	SVR	24.44	17.16	20.94	24.98
peppers	UR	25.33	21.27	22.81	23.61
	SVR	24.22	17.38	18.59	19.22
airplane	UR	24.38	18.08	23.47	23.67
	SVR	21.45	15.28	16.34	18.41
cameraman	UR	7.00	5.23	5.81	6.28
	SVR	6.17	5.02	4.94	5.05
Zelda	UR	26.08	19.70	21.56	24.08
	SVR	24.92	17.97	17.86	20.09
Goldhill	UR	28.25	19.69	20.84	22.78
	SVR	26.53	17.66	16.67	18.61
Mean	UR	26.41	20.03	21.74	23.34
	SVR	24.34	17.51	18.63	19.20

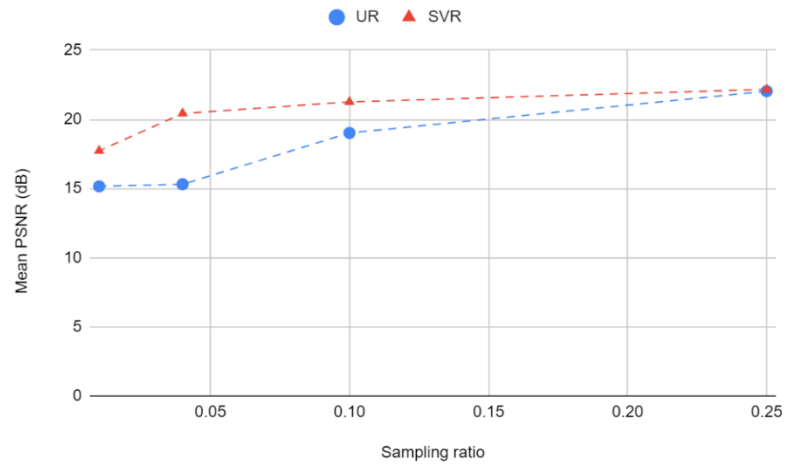


Figure 4.29: The values of mean PSNR (in dB) of images recovered via UR and SVR CPA sensing matrices, without applying BM3D.

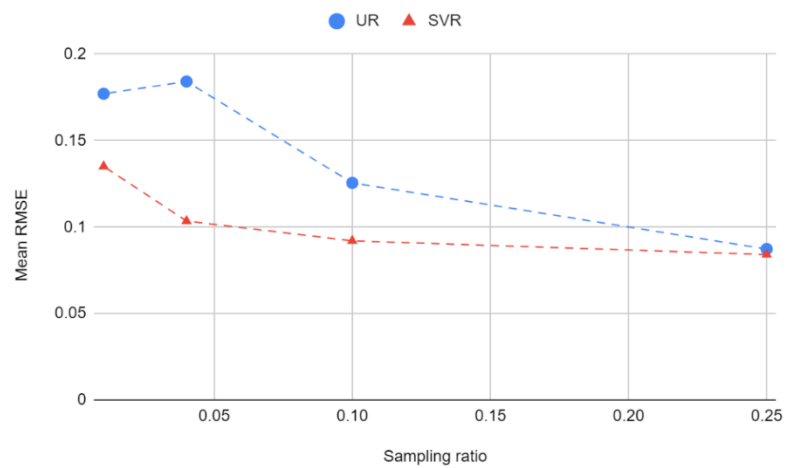


Figure 4.30: The values of mean RMSE of images recovered via UR and SVR CPA sensing matrices, without applying BM3D.

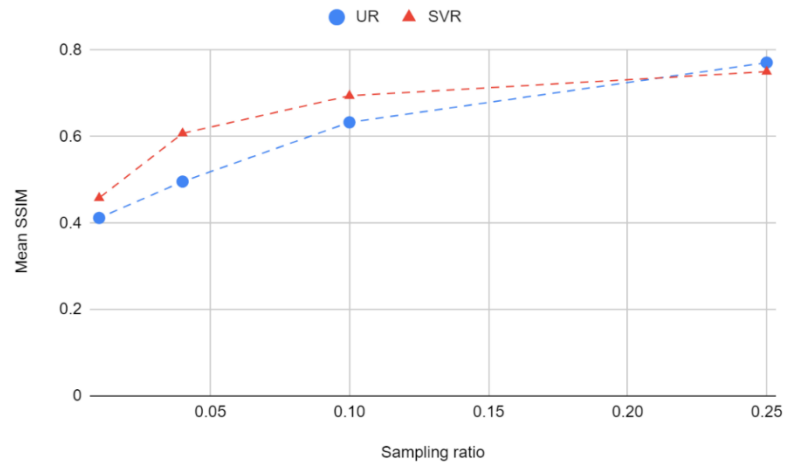


Figure 4.31: The values of mean SSIM of images recovered via UR and SVR CPA sensing matrices, without applying BM3D.

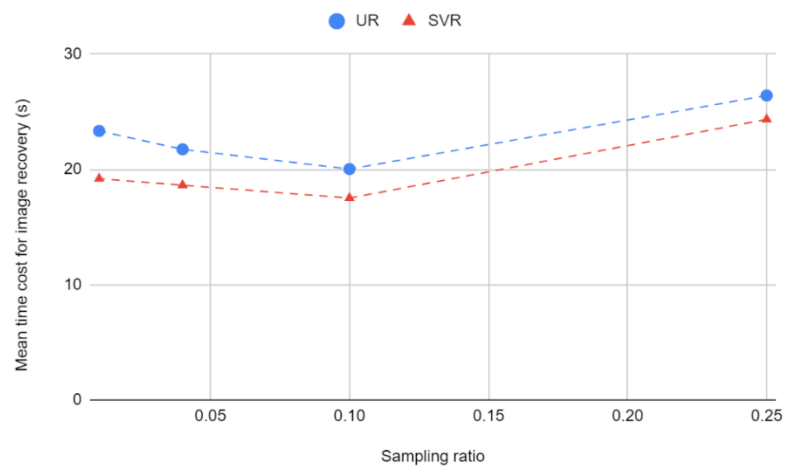


Figure 4.32: The mean time cost for image recovery of images recovered via UR and SVR CPA sensing matrices, without applying BM3D.

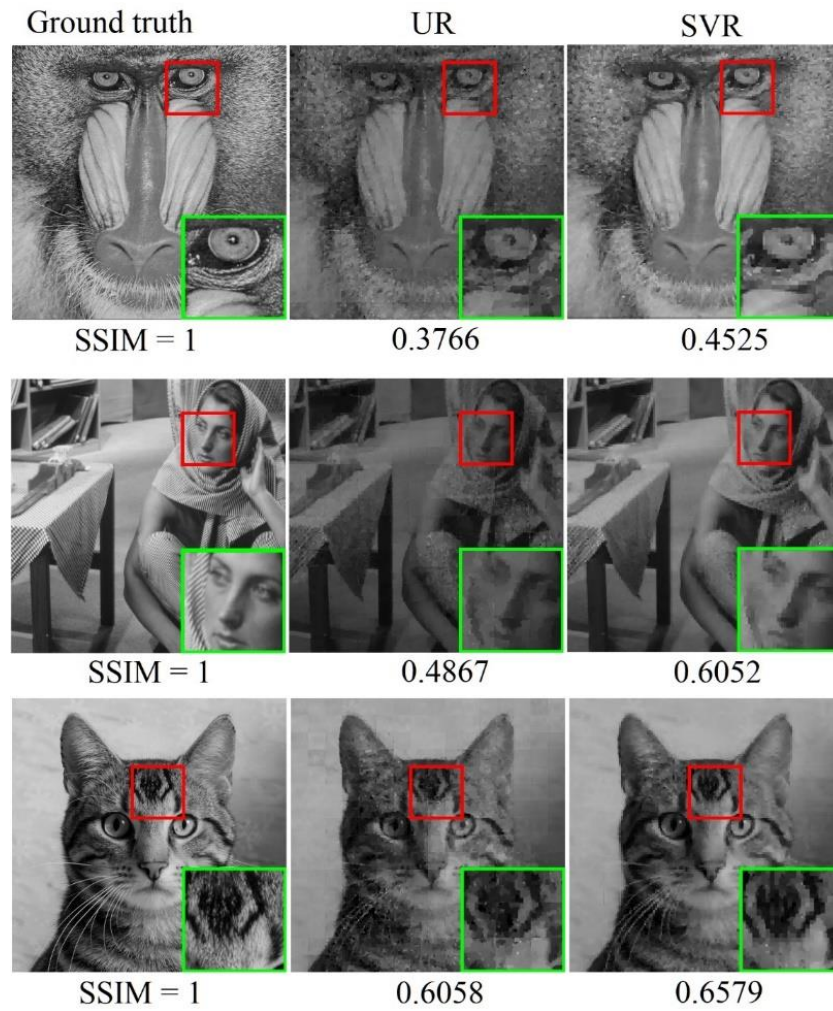


Figure 4.33: (From top to bottom row) The original images “baboon”, “Barbara”, and “cat” and their images recovered via UR and SVR CPA sensing matrices from $SR = 0.1$.

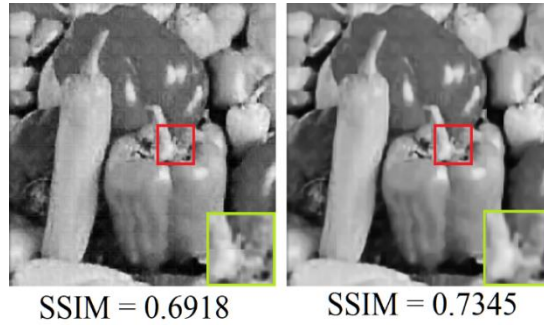


Figure 4.34: (Left) "peppers" recovered via SVR CPA sensing matrices from $SR = 0.04$, without applying BM3D. (Right) "peppers" recovered via SVR CPA sensing matrices from $SR = 0.04$, after applying BM3D. It can be seen that the images became less noisy and the value of SSIM increased after BM3D is applied.

The results shown in Table 4.21 – 4.24 and Figure 4.29 – 4.32 show that the values of the mean PSNR of images recovered via SVR CPA sensing matrices are higher than that recovered via UR CPA sensing matrices for all SR tested, with and without applying BM3D. The values of the mean RMSE of images recovered via SVR CPA sensing matrices are lower than that recovered via UR CPA sensing matrices for all SR tested, with and without applying BM3D. The mean SSIM of images recovered via SVR CPA sensing matrices are higher than that recovered via UR sensing matrices for all SR tested except for $SR = 0.25$, with and without applying BM3D. Further, the mean time cost for image recovery for SVR CPA sensing matrices is lower than that of UR CPA sensing matrices for all SR tested.

From Figure 4.33, it can be observed that the image details of images recovered via SVR CPA sensing matrices are much finer than that recovered

via UR CPA sensing matrices for the same value of SR . Hence, for the same number of measurements, images recovered via SVR CPA sensing matrices have a better image quality than images recovered via UR CPA sensing matrices. This is especially true for smaller numbers of measurements. This indicates that SVR CPA sensing matrices are able to improve the image quality and time efficiency. Hence, it is more superior than UR CPA sensing matrices for block-based approach. Moreover, from Figure 4.34 it can be observed that after applying BM3D, the edges of the image blocks became less prominent and the noise of the image has reduced.

4.3 Block-Based Single-Pixel Imaging with ReconNet

4.3.1 The Network Training Results of ReconNet

Figure 4.35 shows the training and test loss of the network training for each value of SR . The network training time for each value of SR ranges from approximately half a day to one and a half day, with larger value of SR requiring longer training time.

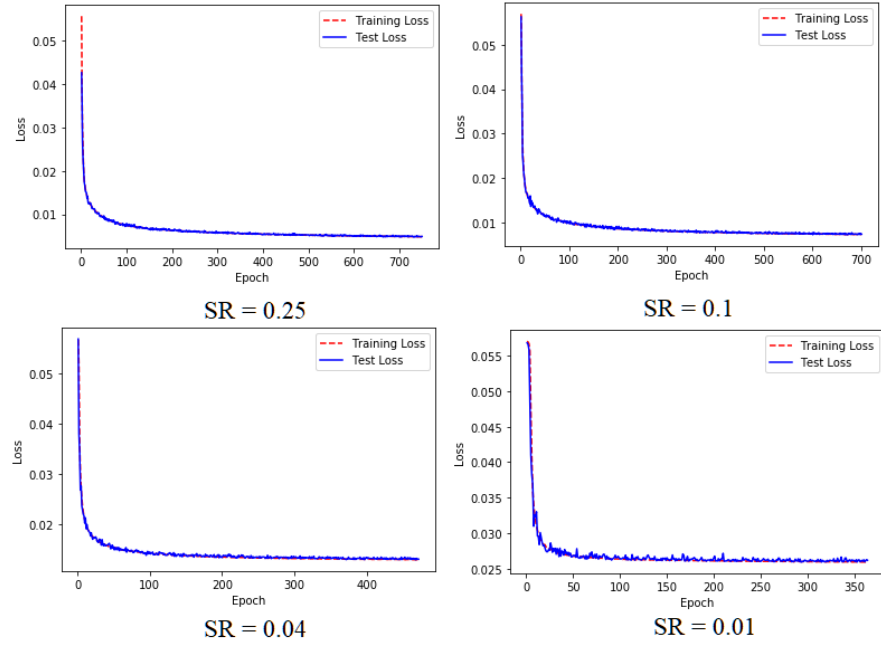


Figure 4.35: The training and test loss of the network training for each value of SR .

4.3.2 Comparison of ReconNet And Other CS Image Recovery Algorithms

In this section the performance and efficiency of ReconNet and other CS image recovery algorithms were studied extensively. The conventional CS image recovery algorithms chosen for comparison are TVAL3, SL0, and 11-magic (Li et al., 2009; Eftekhari et al., 2009; Candes and Romberg, 2005). Table 4.25 – 4.28 show the values of PSNR (in dB), RMSE, SSIM, and time cost for image recovery (in seconds) of each individual image recovered via SVR CPA sensing matrices from $SR \in \{0.25, 0.1, 0.04, 0.01\}$ through ReconNet, TVAL3, SL0, and 11-magic, with and without applying BM3D, for block-based approach. Figure 4.36 – 4.39 show the values of mean PSNR (in dB), RMSE, SSIM, and time cost for image recovery (in seconds) of images recovered via

SVR CPA sensing matrices through ReconNet, TVAL3, SL0, and l1-magic, from $SR \in \{0.25, 0.1, 0.04, 0.01\}$, without applying BM3D, for block-based approach.

Table 4.25: The values of PSNR (in dB) of each individual image recovered via SVR CPA sensing matrices, through ReconNet, TVAL3, SL0, and l1-magic, from $SR \in \{0.25, 0.1, 0.04, 0.01\}$, with and without applying BM3D, for block-based approach. The highest values in each setting is in bold.

Images	Algorithms	PSNR (dB)							
		SR = 0.25		0.1		0.04		0.01	
		w/o BM3D	w/ BM3D	w/o BM3D	w/ BM3D	w/o BM3D	w/ BM3D	w/o BM3D	w/ BM3D
baboon	TVAL3	20.4220	19.9904	20.2632	20.1072	18.7481	18.8346	18.0764	18.2051
	ReconNet	19.6273	19.6948	18.6977	18.7890	17.4561	17.5457	17.5049	17.6129
	SL0	16.8515	17.2373	14.8272	15.7705	14.5413	15.6458	13.5628	14.7284
	l1-magic	16.9264	17.2983	15.6816	16.1727	11.1220	11.2060	13.2778	14.4523
Barbara	TVAL3	17.5522	17.4628	16.9564	16.9383	21.1707	21.4387	18.1059	18.2600
	ReconNet	23.0787	23.3695	22.8419	23.1476	21.9148	22.1712	19.0900	19.2493
	SL0	15.1350	15.3306	13.1806	13.6234	12.9571	13.5055	12.2572	12.8800
	l1-magic	15.3219	15.5814	13.8566	14.0134	11.7020	11.7778	12.1939	12.8424
cat	TVAL3	24.8930	24.6888	19.8230	19.7902	21.0106	21.1109	19.2643	19.4372
	ReconNet	21.4459	21.5121	19.8152	19.8906	18.9134	18.9999	19.6171	19.7744
	SL0	14.9837	15.3813	13.2356	13.8592	12.7937	13.4546	12.0050	12.6763
	l1-magic	15.0574	15.4709	13.1521	13.7694	10.2228	10.2914	11.8889	12.5701
fruits	TVAL3	24.6767	24.7337	23.0320	23.2473	13.5095	13.4945	15.2705	15.3579
	ReconNet	25.7607	25.7661	24.1399	24.2225	21.3264	21.4665	16.8745	16.9767
	SL0	12.8982	12.9792	11.4976	11.7520	11.1338	11.4438	10.4234	10.7757
	l1-magic	12.5133	12.5198	11.5673	11.6195	10.0655	10.0827	10.4271	10.7875
Lenna	TVAL3	25.9046	26.3758	23.5326	23.7905	23.0157	23.3656	20.0064	20.2385
	ReconNet	26.9912	27.3906	26.3149	26.6250	24.8326	25.1774	20.8344	21.0331
	SL0	16.1312	16.5047	14.5372	15.3092	14.1194	14.9724	13.1829	14.0676

	II-magic	16.4366	16.9341	14.7880	15.0332	12.6036	12.7034	13.0610	13.9759
peppers	TVAL3	22.6749	22.8633	22.0697	22.3122	24.0429	24.5062	19.1021	19.3151
	ReconNet	24.8274	25.0282	24.4515	24.6467	23.9509	24.2569	19.9969	20.1823
	SL0	15.6068	16.0817	13.3505	13.9593	13.1731	13.8565	12.2001	12.9199
	II-magic	15.6731	16.1435	13.8513	14.1519	10.7703	10.8478	12.1826	12.9439
airplane	TVAL3	21.0784	21.2475	24.3869	24.6355	21.8945	22.0825	12.3712	12.3916
	ReconNet	21.1710	21.2199	21.3022	21.3485	20.5118	20.6564	15.0838	15.1318
	SL0	12.0226	12.3253	12.5947	13.4551	11.5269	12.2625	11.2958	12.1152
	II-magic	12.6779	13.1064	11.5887	11.8009	6.9720	7.0100	11.2016	12.0041
cameraman	TVAL3	23.6710	23.8155	22.4618	22.6705	20.2102	20.3751	16.5566	16.6425
	ReconNet	19.2982	19.3275	19.4737	19.5335	19.4217	19.5623	15.4981	15.5981
	SL0	14.4747	14.8675	12.3408	12.8919	12.0466	12.6537	11.1873	11.7910
	II-magic	14.5885	15.0099	12.5341	12.8048	11.0959	11.1549	11.0320	11.6445
Zelda	TVAL3	15.7000	15.7433	15.4379	15.4819	15.7146	15.7761	17.9938	18.1390
	ReconNet	17.7725	17.8961	17.2276	17.3380	18.6442	18.8944	15.9636	16.1102
	SL0	15.4041	15.8396	12.2143	12.6456	12.6411	13.2439	11.5390	12.1186
	II-magic	15.3855	15.8308	11.9769	12.4210	12.8502	13.4910	11.2409	11.8553
Goldhill	TVAL3	25.0871	25.0962	24.6382	24.7779	24.9298	25.1070	20.4780	20.6224
	ReconNet	27.0661	27.4428	24.7380	24.8612	23.0977	23.4619	17.5335	17.7035
	SL0	16.0259	16.2637	13.8833	14.3233	13.7995	14.3404	12.8220	13.3784
	II-magic	15.6583	15.7417	14.4608	14.6023	12.0590	12.1310	12.7963	13.3684
Mean	TVAL3	22.1660	22.2017	21.2602	21.3752	20.4247	20.6091	17.7225	17.8609
	ReconNet	22.7039	22.8648	21.9003	22.0403	21.0070	21.2193	17.7997	17.9372
	SL0	14.9534	15.2811	13.1662	13.7590	12.8732	13.5379	12.0475	12.7451
	II-magic	15.0239	15.3637	13.3457	13.6389	10.9463	11.0696	11.9302	12.6444

Table 4.26: The values of RMSE of each individual image recovered via SVR CPA sensing matrices, through ReconNet, TVAL3, SL0, and l1-magic, from $SR \in \{0.25, 0.1, 0.04, 0.01\}$, with and without applying BM3D, for block-based approach. The lowest value in each setting is in bold.

Images	Algorithms	RMSE							
		SR = 0.25		0.1		0.04		0.01	
		w/o BM3D	w/ BM3D	w/o BM3D	w/ BM3D	w/o BM3D	w/ BM3D	w/o BM3D	w/ BM3D
baboon	TVAL3	0.0953	0.1001	0.0970	0.0988	0.1155	0.1144	0.1248	0.1230
	ReconNet	0.1044	0.1036	0.1162	0.1150	0.1340	0.1327	0.1333	0.1316
	SL0	0.1437	0.1374	0.1814	0.1627	0.1875	0.1875	0.2098	0.1835
	l1-magic	0.1425	0.1365	0.1644	0.1554	0.2779	0.2752	0.2168	0.1894
Barbara	TVAL3	0.1326	0.1339	0.1420	0.1423	0.0874	0.0847	0.1244	0.1222
	ReconNet	0.0702	0.0702	0.0721	0.0696	0.0802	0.0779	0.1110	0.1090
	SL0	0.1751	0.1712	0.2193	0.2084	0.2250	0.2112	0.2439	0.2270
	l1-magic	0.1714	0.1663	0.2028	0.1992	0.2600	0.2577	0.2456	0.2280
cat	TVAL3	0.0569	0.0583	0.1021	0.1024	0.0890	0.0880	0.1088	0.1067
	ReconNet	0.0847	0.0840	0.1022	0.1013	0.1133	0.1122	0.1045	0.1026
	SL0	0.1782	0.1702	0.2179	0.2028	0.2293	0.2125	0.2510	0.2324
	l1-magic	0.1767	0.1684	0.2200	0.2049	0.3082	0.3058	0.2544	0.2352
fruits	TVAL3	0.0584	0.0580	0.0705	0.0688	0.2111	0.2115	0.1724	0.1706
	ReconNet	0.0515	0.0515	0.0621	0.0615	0.0858	0.0845	0.1433	0.1416
	SL0	0.2265	0.2244	0.2661	0.2585	0.2775	0.2678	0.3012	0.2892
	l1-magic	0.2368	0.2366	0.2640	0.2624	0.3139	0.3132	0.3011	0.2888
Lenna	TVAL3	0.0507	0.0480	0.0666	0.0646	0.0707	0.0679	0.0999	0.0973
	ReconNet	0.0447	0.0427	0.0483	0.0466	0.0573	0.0551	0.0908	0.0888
	SL0	0.1561	0.1495	0.1876	0.1716	0.1968	0.1784	0.2192	0.1980
	l1-magic	0.1507	0.1423	0.1822	0.1772	0.2343	0.2316	0.2223	0.2001
peppers	TVAL3	0.0735	0.0719	0.0788	0.0766	0.0628	0.0595	0.1109	0.1082
	ReconNet	0.0574	0.0561	0.0599	0.0586	0.0635	0.0613	0.1000	0.0979
	SL0	0.1658	0.1570	0.2150	0.2005	0.2195	0.2028	0.2455	0.2259
	l1-magic	0.1646	0.1559	0.2030	0.1961	0.2894	0.2868	0.2460	0.2253
airplane	TVAL3	0.0883	0.0866	0.0603	0.0586	0.0804	0.0787	0.2407	0.2401

	ReconNet	0.0874	0.0869	0.0861	0.0856	0.0943	0.0927	0.1761	0.1751
	SL0	0.2505	0.2420	0.2346	0.2124	0.2653	0.2437	0.2724	0.2479
	l1-magic	0.2323	0.2211	0.2634	0.2570	0.4481	0.4462	0.2754	0.2511
cameraman	TVAL3	0.0655	0.0644	0.0753	0.0735	0.0976	0.0958	0.1487	0.1472
	ReconNet	0.1084	0.1081	0.1062	0.1055	0.1069	0.1052	0.1679	0.1660
	SL0	0.1889	0.1806	0.2415	0.2267	0.2498	0.2330	0.2758	0.2573
Zelda	l1-magic	0.1865	0.1776	0.2362	0.2290	0.2787	0.2769	0.2808	0.2617
	TVAL3	0.1641	0.1632	0.1691	0.1682	0.1638	0.1626	0.1260	0.1239
	ReconNet	0.1292	0.1274	0.1376	0.1359	0.1169	0.1136	0.1592	0.1565
Goldhill	SL0	0.1697	0.1614	0.2451	0.2332	0.2333	0.2177	0.2649	0.2478
	l1-magic	0.1701	0.1616	0.2519	0.2393	0.2278	0.2116	0.2741	0.2554
	TVAL3	0.0557	0.0556	0.0586	0.0577	0.0567	0.0555	0.0946	0.0931
Mean	ReconNet	0.0443	0.0424	0.0580	0.0571	0.0700	0.0671	0.1328	0.1303
	SL0	0.1580	0.1537	0.2022	0.1922	0.2042	0.1919	0.2285	0.2143
	l1-magic	0.1648	0.1633	0.1892	0.1862	0.2495	0.2474	0.2292	0.2146
Mean	TVAL3	0.0841	0.0840	0.0920	0.0912	0.1035	0.1019	0.1351	0.1332
	ReconNet	0.0782	0.0773	0.0849	0.0837	0.0922	0.0902	0.1319	0.1300
	SL0	0.1813	0.1747	0.2211	0.2069	0.2288	0.2146	0.2512	0.2323
Mean	l1-magic	0.1796	0.1730	0.2177	0.2107	0.2888	0.2852	0.2546	0.2350

Table 4.27: The values of SSIM of each individual image recovered via SVR CPA sensing matrices, through ReconNet, TVAL3, SL0, and l1-magic, from $SR \in \{0.25, 0.1, 0.04, 0.01\}$, with and without applying BM3D, for block-based approach. The highest value in each setting is in bold.

Images	Algorithms	SSIM							
		SR = 0.25		0.1		0.04		0.01	
		w/o BM3D	w/ BM3D	w/o BM3D	w/ BM3D	w/o BM3D	w/ BM3D	w/o BM3D	w/ BM3D
baboon	TVAL3	0.5392	0.4302	0.4525	0.3873	0.3584	0.3402	0.2500	0.2557
	ReconNet	0.5625	0.5246	0.4757	0.4488	0.3637	0.3580	0.2415	0.2415
	SL0	0.2743	0.2714	0.1456	0.2201	0.0983	0.1732	0.0617	0.1249
	l1-magic	0.2793	0.2731	0.1673	0.2312	0.1391	0.1917	0.0559	0.1075
Barbara	TVAL3	0.6657	0.6819	0.6395	0.6572	0.5643	0.5951	0.4075	0.4318
	ReconNet	0.6522	0.6875	0.6181	0.6585	0.5492	0.5942	0.3939	0.4239

	SL0	0.3575	0.4400	0.2004	0.3589	0.1434	0.2982	0.0988	0.2349
	11-magic	0.3589	0.4478	0.2632	0.3797	0.2524	0.3609	0.0915	0.2188
cat	TVAL3	0.7344	0.6931	0.6579	0.6374	0.5862	0.5956	0.4641	0.4901
	ReconNet	0.6961	0.7093	0.6615	0.6711	0.5919	0.6130	0.4190	0.4467
	SL0	0.3475	0.5011	0.1761	0.3467	0.1219	0.2518	0.0819	0.1891
	11-magic	0.3517	0.5017	0.1748	0.3423	0.2133	0.3634	0.0753	0.1681
fruits	TVAL3	0.7972	0.7713	0.7149	0.7141	0.5194	0.5258	0.3648	0.3941
	ReconNet	0.8003	0.8012	0.7316	0.7415	0.5923	0.6180	0.3472	0.3680
	SL0	0.3445	0.4084	0.1959	0.3134	0.1416	0.2507	0.0944	0.1920
	11-magic	0.3400	0.3865	0.2409	0.3335	0.2457	0.3091	0.0908	0.1855
Lenna	TVAL3	0.8228	0.8208	0.7878	0.7878	0.7062	0.7372	0.5430	0.5823
	ReconNet	0.8110	0.8416	0.7895	0.8153	0.7110	0.7483	0.5454	0.5789
	SL0	0.4328	0.6198	0.2259	0.4904	0.1593	0.3911	0.1064	0.2952
	11-magic	0.4309	0.6274	0.3305	0.5511	0.3226	0.5178	0.0970	0.2612
peppers	TVAL3	0.8038	0.8046	0.7679	0.7854	0.6986	0.7330	0.5116	0.5497
	ReconNet	0.7906	0.8208	0.7713	0.7992	0.6918	0.7345	0.5073	0.5421
	SL0	0.4001	0.5941	0.2134	0.4565	0.1554	0.3713	0.1034	0.2768
	11-magic	0.4060	0.5960	0.2797	0.5083	0.2853	0.4548	0.0947	0.2484
airplane	TVAL3	0.8299	0.8282	0.7895	0.7936	0.7009	0.7254	0.5399	0.5642
	ReconNet	0.7695	0.8290	0.7318	0.7910	0.6338	0.7039	0.5210	0.5689
	SL0	0.3656	0.6012	0.1663	0.3889	0.1043	0.2398	0.0646	0.1555
	11-magic	0.3642	0.6103	0.2539	0.5293	0.2016	0.3975	0.0607	0.1364
cameraman	TVAL3	0.7804	0.7559	0.7280	0.7230	0.6500	0.6615	0.4953	0.5177
	ReconNet	0.6721	0.7169	0.6355	0.6855	0.5293	0.5937	0.3576	0.3958
	SL0	0.3312	0.5066	0.1603	0.3231	0.1103	0.2239	0.0759	0.1581
	11-magic	0.3329	0.5080	0.2046	0.4028	0.2259	0.3904	0.0683	0.1282
Zelda	TVAL3	0.8045	0.8191	0.7687	0.7924	0.7120	0.7541	0.5717	0.6179
	ReconNet	0.6959	0.8036	0.6786	0.7795	0.5792	0.6852	0.3686	0.4099
	SL0	0.4226	0.6387	0.2184	0.4780	0.1548	0.3781	0.1052	0.2844
	11-magic	0.4267	0.6430	0.2111	0.4627	0.1548	0.3765	0.0869	0.2219
Goldhill	TVAL3	0.7298	0.6931	0.6730	0.6582	0.5849	0.5891	0.4425	0.4613
	ReconNet	0.6850	0.7058	0.6247	0.6436	0.5135	0.5586	0.3315	0.3564
	SL0	0.3916	0.4652	0.2344	0.3922	0.1763	0.3365	0.1255	0.2799
	11-magic	0.4027	0.4497	0.3064	0.4148	0.2830	0.3821	0.1191	0.2696
Mean	TVAL3	0.7508	0.7298	0.6980	0.6936	0.6081	0.6257	0.4590	0.4865

	ReconNet	0.7135	0.7440	0.6718	0.7034	0.5756	0.6207	0.4033	0.4332
	SL0	0.3668	0.5046	0.1937	0.3768	0.1366	0.2915	0.0918	0.2191
	l1-magic	0.3693	0.5043	0.2432	0.4156	0.2324	0.3744	0.0840	0.1946

Table 4.28: The time cost for image recovery (in seconds) for each individual image recovered via SVR CPA sensing matrices through ReconNet, TVAL3, SL0, and l1-magic, from $SR \in \{0.25, 0.1, 0.04, 0.01\}$, without applying BM3D, for block-based approach. The lowest time cost in each setting is in bold.

Images	Algorithms	Time cost for image recovery (s)			
		SR = 0.25	0.1	0.04	0.01
baboon	TVAL3	27.27	23.14	16.72	24.23
	ReconNet	0.27	0.27	0.30	0.32
	SL0	21.13	5.48	2.50	0.88
	l1-magic	133.36	90.59	77.81	78.22
Barbara	TVAL3	29.36	18.17	22.95	19.19
	ReconNet	0.30	0.33	0.33	0.31
	SL0	20.95	5.33	2.50	0.89
	l1-magic	142.19	98.23	83.78	77.88
cat	TVAL3	26.47	18.36	24.05	19.09
	ReconNet	0.31	0.32	0.29	0.38
	SL0	20.53	5.28	2.59	0.91
	l1-magic	148.53	96.41	87.67	82.23
fruits	TVAL3	32.61	25.02	27.28	23.11
	ReconNet	0.34	0.32	0.28	0.31
	SL0	21.81	5.09	2.50	0.91
	l1-magic	150.16	99.00	84.41	71.45
Lenna	TVAL3	24.44	17.16	20.94	24.98
	ReconNet	0.28	0.32	0.27	0.31
	SL0	20.45	5.63	2.77	0.88
	l1-magic	142.70	95.53	84.80	80.23
peppers	TVAL3	24.22	17.38	18.59	19.22
	ReconNet	0.30	0.32	0.31	0.31

	SL0	20.36	5.69	2.59	0.86
	11-magic	147.94	101.19	84.36	82.20
airplane	TVAL3	21.45	15.28	16.34	18.41
	ReconNet	0.39	0.41	0.28	0.33
	SL0	23.52	5.95	2.94	0.89
	11-magic	169.78	116.27	104.34	92.20
cameraman	TVAL3	6.17	5.02	4.94	5.05
	ReconNet	0.17	0.15	0.15	0.13
	SL0	5.33	1.41	0.67	0.22
	11-magic	48.09	23.59	19.88	18.64
Zelda	TVAL3	24.92	17.97	17.86	20.09
	ReconNet	0.31	0.26	0.28	0.27
	SL0	27.64	5.08	2.64	0.88
	11-magic	155.27	113.67	93.50	71.58
Goldhill	TVAL3	26.53	17.66	16.67	18.61
	ReconNet	0.29	0.27	0.26	0.25
	SL0	30.91	5.33	2.52	0.88
	11-magic	168.78	94.73	85.08	70.45
Mean	TVAL3	24.34	17.51	18.63	19.20
	ReconNet	0.30	0.30	0.28	0.29
	SL0	21.26	5.03	2.42	0.82
	11-magic	140.68	92.92	80.56	72.51

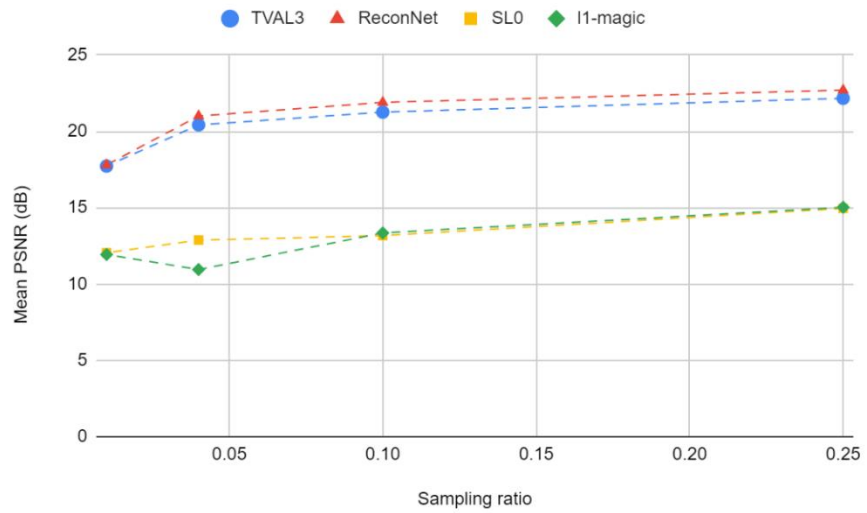


Figure 4.36: The values of mean PSNR (in dB) of images recovered via SVR CPA sensing matrices through ReconNet, TVAL3, SL0, and l1-magic, from $SR \in \{0.25, 0.1, 0.04, 0.01\}$, without applying BM3D, for block-based approach.

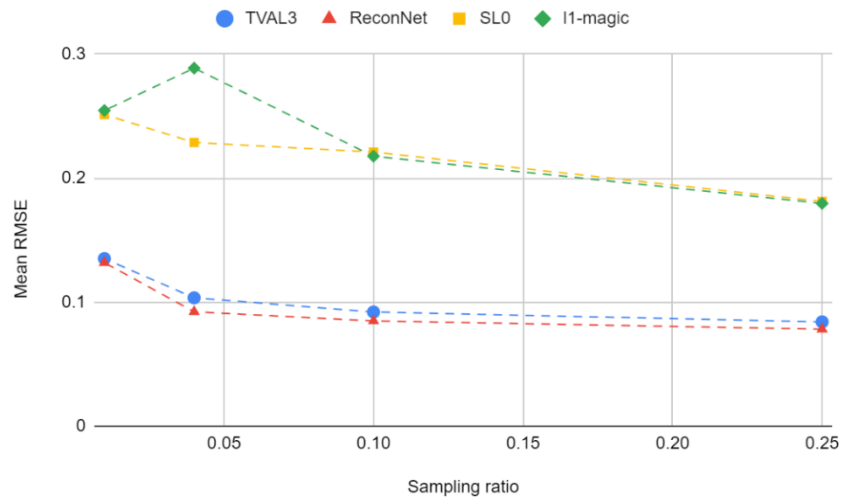


Figure 4.37: The values of mean RMSE of images recovered via SVR CPA sensing matrices through ReconNet, TVAL3, SL0, and l1-magic, from $SR \in \{0.25, 0.1, 0.04, 0.01\}$, without applying BM3D, for block-based approach.

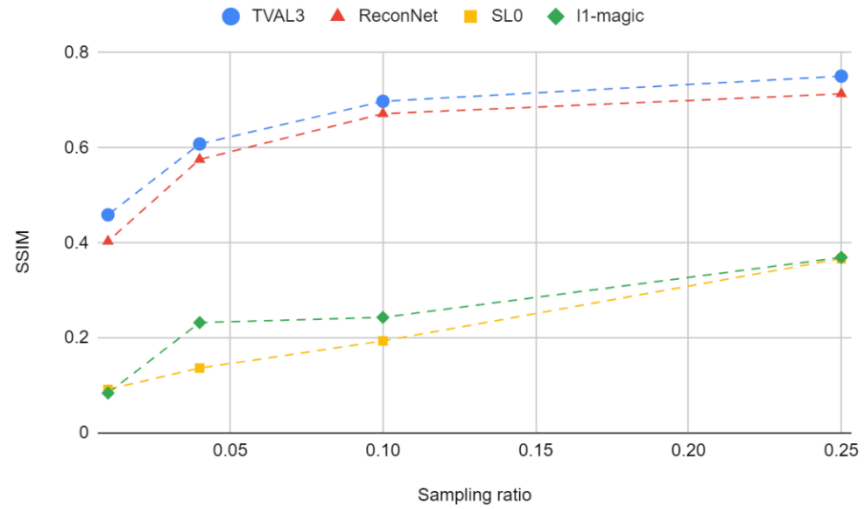


Figure 4.38: The values of mean SSIM of images recovered via SVR CPA sensing matrices through ReconNet, TVAL3, SL0, and l1-magic, from $SR \in \{0.25, 0.1, 0.04, 0.01\}$, without applying BM3D, for block-based approach.

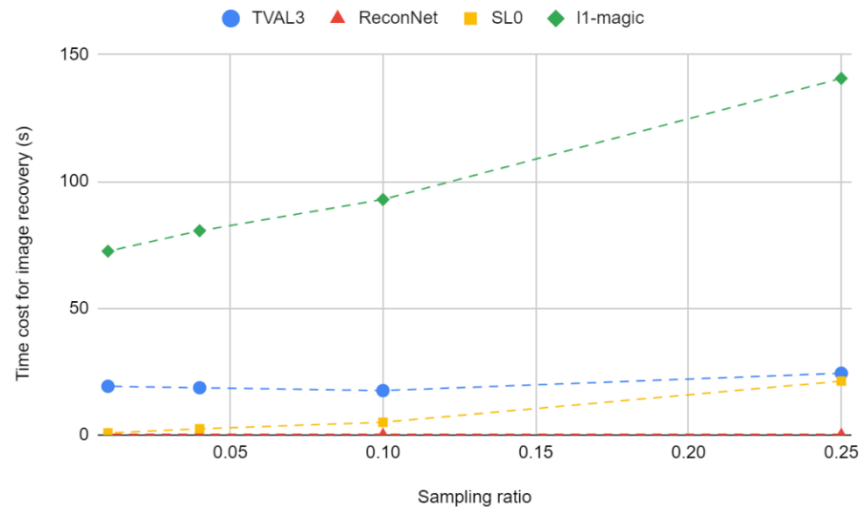


Figure 4.39: The mean time cost for image recovery of images recovered via SVR CPA sensing matrices through ReconNet, TVAL3, SL0, and l1-magic, from $SR \in \{0.25, 0.1, 0.04, 0.01\}$, without applying BM3D, for block-based approach.

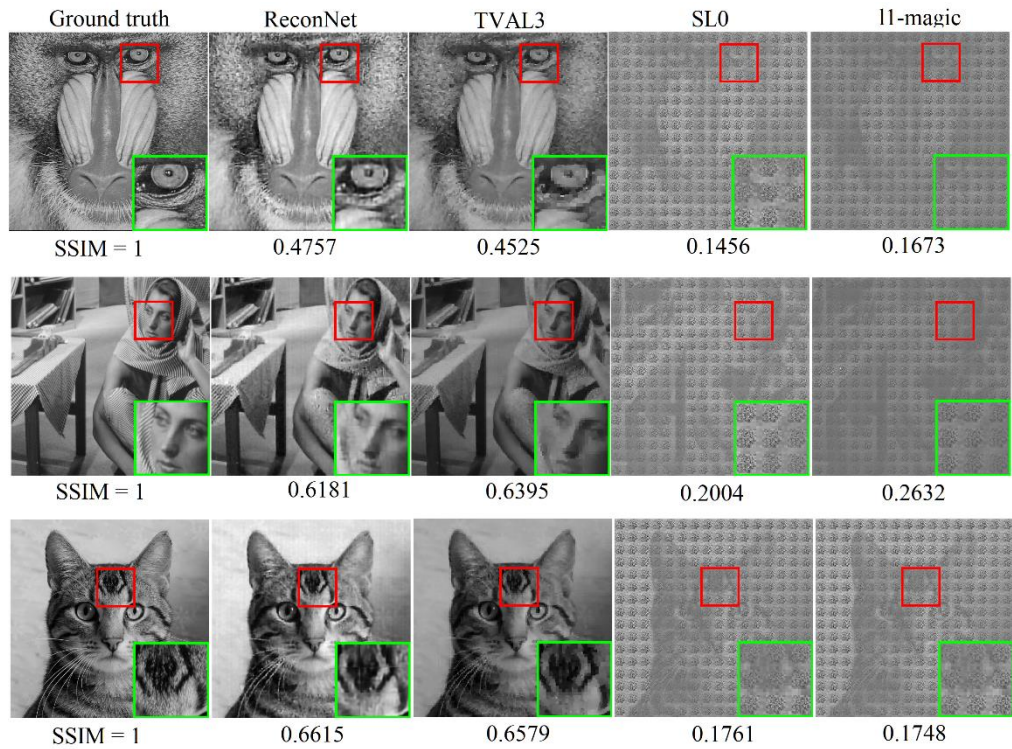


Figure 4.40: (From top to bottom row) The original images “baboon”, “Barbara”, and “cat” and their images recovered via SVR CPA sensing matrices through ReconNet, TVAL3, SL0, and l1-magic from $SR = 0.1$, without applying BM3D, for block-based approach. All images are shown with their respective values of SSIM.

From the results, it can be seen that the value of the mean PSNR (in dB) is the highest for images recovered via SVR CPA sensing matrices through ReconNet for $SR \in \{0.25, 0.1, 0.04, 0.01\}$. The mean value of RMSE is the lowest for images recovered via SVR CPA sensing matrices through ReconNet for $SR \in \{0.25, 0.1, 0.04, 0.01\}$. The mean values of SSIM for images recovered via SVR CPA sensing matrices through ReconNet and TVAL3 are the highest and comparable for $SR \in \{0.25, 0.1, 0.04, 0.01\}$. The mean time cost for image recovery (in seconds) is the shortest for images recovered via SVR

CPA sensing matrices through ReconNet for $SR \in \{0.25, 0.1, 0.04, 0.01\}$. Visually, Figure 4.40 shows that for the same value of SR , the image details of images recovered via SVR CPA sensing matrices through ReconNet are clearer than that of images recovered through TVAL3, SL0, and l1-magic.

The results indicate that ReconNet is able to improve the image quality. Furthermore, the mean time cost for image recovery for needed for ReconNet is significantly less than the conventional CS image recovery algorithms. Since the objectives of this project are to keep the number of measurements as low as possible while maintaining a higher image quality, the results show that ReconNet is a more favorable choice for CS image recovery algorithms than the conventional CS image recovery algorithms.

4.4 Image Encryption Scheme with CS Via CPA Sensing Matrices

Aside from imaging applications, single-pixel imaging can also be applied in image encryption and decryption. In image encryption, the sensing matrices act as the encryption keys. Then, the same sensing matrices can be used later for image decryption. Since conventionally single-pixel imaging utilizes randomly generated sensing matrices, the random nature of the keys makes it nearly impossible for the eavesdroppers to obtain information pertaining the keys. However, storing the randomly generated sensing matrices are more difficult as one has to save the entire sensing matrices.

This section demonstrates the image encryption and decryption scheme by applying SVR CPA sensing matrices. In the scenario of image encryption with CPA sensing matrices, the seeds z_0 that are used to generate CPA sensing matrices are acting as the keys which are used in the encryption and decryption

processes. For demonstration, $z_0 = 0.19$ was chosen as the right key. Five SVR CPA sensing matrices were generated with $z_0 \in \{0.19, 0.39, 0.59, 0.79, 0.99\}$. Then, the CS measurements of “peppers” were obtained with the SVR CPA sensing matrix generated with $z_0 = 0.19$. Finally, the image was recovered from the CS measurements of “peppers” via different keys z_0 , through TVAL3. Figure 4.41 shows the recovered images via SVR CPA sensing matrices generated with different keys z_0 . From Figure 4.41, it can be seen that except for the right key $z_0 = 0.19$, all other keys failed to recover the original image. Therefore, the secrecy of the right key guarantees the safety of the image. CPA is more suitable for image encryption scheme than the other sensing matrices as only the values of the keys are needed to be stored, instead of the entire sensing matrices.

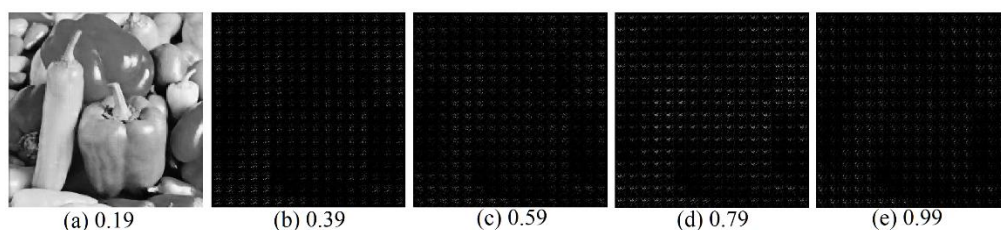


Figure 4.41: “peppers” recovered via SVR CPA sensing matrices generated with $z_0 =$ (a) 0.19 (b) 0.39 (c) 0.59 (d) 0.79 (e) 0.99. It can be seen that except for the right key $z_0 = 0.19$, all other keys failed to recover the original image.

CHAPTER 5

CONCLUSIONS AND FUTURE WORKS

5.1 Conclusion

The development of single-pixel imaging has provided a much simpler and lower cost imaging system architecture, as well as a mean to acquire images in wavelengths at the invisible light spectrum and conditions in which photons are scarce. Furthermore, compressed sensing (CS) makes it possible to compress and acquire the significant image data at the same time in the image acquisition process.

This project proposes a design of programmable SVR sensing patterns used in the CS image acquisition and recovery process. In the numerical experiments, the performance of different spatially-variant resolution (SVR) sensing matrices were evaluated. The results show that SVR chaotic patterns array (CPA) sensing matrices is able to give better image quality than SVR random binary patterns (RBP) and Hadamard patterns (HMP) sensing matrices. Furthermore, the results show that SVR CPA sensing matrices are able to give better image quality at lower sampling rate. Accordingly, the performance of proposed SVR method is compared to the conventional uniform resolution (UR) sensing patterns. The results show that the mean time cost required for image recovery using SVR CPA sensing matrices is consistently lower than UR CPA sensing matrices. In conclusion, CPA is more suitable than RBP and HMP for the proposed SVR approach. Furthermore, SVR sensing patterns are more favorable than the conventional UR sensing patterns in CS for situations where

the numbers of measurements are small, yet they are able to give a better image quality. The improved efficiency also indicates the potential of SVR sensing patterns in practical cases as compared to UR sensing patterns.

In order to further increase the computational efficiency in the image recovery process of large images, the proposed SVR sensing patterns is applied to block-based CS approach. In the numerical experiments, the performance of block-based CS approach using various sensing matrices were analyzed. Similar to the findings earlier, the results indicate that the quality of images recovered via SVR CPA sensing matrices is better than SVR RBP and HMP sensing matrices. Moreover, the results show that the quality of images recovered via SVR CPA sensing matrices is higher than UR CPA sensing matrices when the number of measurements is small. The time cost required for image recovery using SVR CPA sensing matrices is also lower than UR CPA sensing matrices. Therefore, in agreement with the results earlier, CPA is more suitable for the proposed SVR approach than RBP and HMP. Furthermore, SVR CPA sensing matrices are able to improve the image quality and time efficiency in block-based CS approach.

There exists plenty of conventional iterative CS image recovery algorithms. However, iterative CS image recovery algorithms are typically time consuming. Recently, convolutional neural networks (CNN) such as ReconNet has been applied for image recovery tasks as non-iterative CS image recovery algorithms. In this project, the performance of ReconNet was compared to the conventional CS image recovery algorithms. The results show that with ReconNet, the recovered images have a comparable image quality to that of TVAL3. However, ReconNet reduces the image recovery time significantly as

compared to TVAL3. Hence, it can be concluded that ReconNet is more time-efficient with minimal image quality trade-off and it is more suitable for practical cases than TVAL3.

In conclusion, the proposed SVR approach in this project is more favorable for basic visual tasks and most importantly it requires less measurement. The results also demonstrated that CNN outperforms the conventional CS image recovery algorithms which is particularly suitable for practical cases.

5.2 Recommendations for Future Works

This project has shown that the proposed methods are able to improve the image quality and time efficiency. For future works, it is good to focus on the development of practical application based on the proposed methods. The practicality of the proposed method can be studied from different perspectives involving the external factors and further improvement can be considered.

Next, it is also worthwhile to look into the investigation and fine tuning of the CNN network parameters for possible result improvement in CS image recovery. Due to time constraint, extensive evaluation of the network parameters was not performed for this project. The fine tuning of the network parameters could potentially improve the quality of the recovered images.

Furthermore, it is possible to expand the study into other deep learning models (WDLReconNet, CombNet, DR²-Net, and more) and provide an extensive comparison. A study of the effectiveness of the proposed methods using various deep learning models is interesting and could potentially yields fruitful results.

REFERENCES

- Abouelnaga, Y., Ali, O.S., Rady, H. and Moustafa, M., 2016. CIFAR-10: KNN-based Ensemble of Classifiers. s.l., *Proceed in International Conference on Computational Science and Computational Intelligence (CSCI)*, pp. 1192-1195.
- Abo-Zahhad, M.M., Hussein, A.I. and Mohamed, A.M., 2015. Compressive sensing algorithms for signal processing applications: A survey. *International Journal of Communications, Network and System Sciences*, 8(6), pp. 197.
- Acharya, U. R. et al., 2017. A deep convolutional neural network model to classify heartbeats. *Computers in Biology and Medicine*, 89, pp. 389-396.
- Adler, A., Boubilil, D., Elad, M. and Zibulevsky, M., 2016. A deep learning approach to block-based compressed sensing of images. *arXiv preprint arXiv:1606.01519*.
- Albawi, S., Mohammed, T. A., and Al-Zawi, S., 2017. Understanding of a convolutional neural network. s.l., *Proceedings of the International Conference on Engineering and Technology (ICET)*, pp. 1-6.
- Baraniuk, R., Davenport, M., DeVore, R. and Wakin, M., 2008. A simple proof of the restricted isometry property for random matrices. *Constructive Approximation*, 28(3), pp. 253-263.

Baraniuk, R. G., 2007. Compressive sensing [lecture notes]. *IEEE Signal Processing Magazine*, 24(4), pp. 118-121.

Bo, L., Lu, H., Lu, Y., Meng, J. and Wang, W., 2017. FompNet: Compressive sensing reconstruction with deep learning over wireless fading channels. s.l., *Proceedings of the 9th International Conference on Wireless Communications and Signal Processing (WCSP)*, pp. 1-6.

Candes, E. J., and Tao, T., 2006. Near-optimal signal recovery from random projections: Universal encoding strategies? *IEEE transactions on information theory*, 52(12), pp. 5406-5425.

Candes, E. J., Romberg, J. K., and Tao, T., 2006. Stable signal recovery from incomplete and inaccurate measurements. *Communications on Pure and Applied Mathematics: A Journal Issued by the Courant Institute of Mathematical Sciences*, 59(8), pp. 1207-1223.

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

Candes, E., and Romberg, J., 2005. *l1-magic: Recovery of sparse signals via convex programming*. [Online] Available at: <http://www.acm.caltech.edu/l1magic/downloads/l1magic> [Accessed 1 May 2019].

Candès, E. J., 2006. Compressive sampling. s.l., *Proceedings of the international congress of mathematicians*, pp. 1433-1452.

Candes, E. J., 2008. The restricted isometry property and its implications for compressed sensing. *Comptes rendus mathematique*, 346(9-10), pp. 589-592.

Cohen, A., Dahmen, W. and DeVore, R., 2009. Compressed sensing and best k -term approximation. *Journal of the American Mathematical Society*, 22(1), pp. 211-231.

DeVore, R. A., 2007. Deterministic constructions of compressed sensing matrices. *Journal of complexity*, 23(4-6), pp. 918-925.

Donoho, D. L., and Tanner, J., 2005. Sparse nonnegative solution of underdetermined linear equations by linear programming. *Proceedings of the national academy of sciences*, 102(27), pp. 9446-9451.

Donoho, D. L., 2006. Compressed Sensing. *IEEE Transaction On Information Theory*, 52(4), pp. 1289-1306.

Duarte, M. F. et al., 2008. Single-pixel imaging via compressive sampling. *IEEE Signal Processing Magazine*, 25(2), pp. 83-91.

Edgar, M.P., Gibson, G.M. and Padgett, M.J., 2019. Principles and prospects for single-pixel imaging. *Nature Photonics*, 13(1), pp. 13-20.

Eftekhari, A., Babaie-Zadeh, M., Jutten, C., and Moghaddam, H. A., 2009. Robust-SL0 for stable sparse representation in noisy settings. s.l., *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing*, pp. 3433-3436.

Gan, H. et al., 2019. Chaotic Pattern Array for Single-Pixel Imaging. *Electronics*, 8(5), pp. 536.

Gan, L., 2007. Block compressed sensing of natural images. s.l., *15th International conference on digital signal processing*, pp. 403-406.

Kulkarni, K., Lohit, S., Turaga, P., Kerviche, R., and Ashok, A., 2016. Reconnet: Non-iterative reconstruction of images from compressively sensed measurements. s.l., *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 449-458.

Lawrence, S., Giles, C.L., Tsoi, A.C. and Back, A.D., 1997. Face recognition: A convolutional neural-network approach. *IEEE Transactions on Neural Networks*, 8(1), pp. 98-113.

Li, C., Yin, W., and Zhang, Y., 2009. User's guide for TVAL3: TV minimization by augmented lagrangian and alternating direction algorithms. *CAAM report*, 20(46-47), pp. 4.

Lu, H., and Bo, L., 2019. WDLReconNet: Compressive Sensing Reconstruction With Deep Learning Over Wireless Fading Channels. *IEEE Access*, 7, pp. 24440-24451.

Lu, W., Dai, T. and Xia, S.T., 2017. Binary matrices for compressed sensing. *IEEE Transactions on Signal Processing*, 66(1), pp. 77-85.

Meenakshi, S. B., 2015. A Survey of compressive sensing based Greedy pursuit reconstruction algorithms. *Int. J. of Image Graphics and Signal Processing*, 7(10), pp. 1-10.

Metzler, C.A., Mousavi, A. and Baraniuk, R.G., 2017. Learned D-AMP: Principled neural network based compressive image recovery. *arXiv preprint arXiv:1704.06625*.

Mousavi, A., Patel, A.B. and Baraniuk, R.G., 2015. A deep learning approach to structured signal recovery. s.l., *Proceedings of 53rd annual allerton conference on communication, control, and computing (Allerton)*, pp. 1336-1343.

Phillips, D. B. et al., 2017. Adaptive foveated single-pixel imaging with dynamic supersampling. *Science advances*, 3(4), pp. e1601782.

Rani, M., Dhok, S.B. and Deshmukh, R.B., 2018. A systematic review of compressive sensing: Concepts, implementations and applications. *IEEE Access*, 6, pp. 4875-4894.

Sainath, T.N., Mohamed, A.R., Kingsbury, B. and Ramabhadran, B., 2013. Deep convolutional neural networks for LVCSR. s.l., *Proceedings of the IEEE international conference on acoustics, speech and signal processing*, pp. 8614-8618.

Shi, D. et al., 2015. Enhancing resolution of single-pixel imaging system. *Optical Review*, 22(5), pp. 802-808.

Shin, Z. Y., Lin, H. S., Chai, T. Y., Wang, X., and Chua, S. Y, 2019. Programmable Single-Pixel Imaging. s.l., *Proceedings of the 13th International Conference on Sensing Technology (ICST)*, pp. 1-6.

Shin, Z., Lin, H.S., Chai, T.Y., Wang, X. and Chua, S.Y., 2021. Programmable spatially variant single-pixel imaging based on compressive sensing. *Journal of Electronic Imaging*, 30(2), pp. 021004.

Sun, M.J., Meng, L.T., Edgar, M.P., Padgett, M.J. and Radwell, N., 2017. A Russian Dolls ordering of the Hadamard basis for compressive single-pixel imaging. *Scientific Report*, 7(1), pp. 1-7.

Vujović, S., Daković, M. and Stanković, L., 2014. Comparison of the L1-magic and the Gradient Algorithm for Sparse Signals Reconstruction. s.l., *Proceedings of the 22nd Telecommunications Forum Telfor (TELFOR)*, pp. 577-580.

Yamashita, R., Nishio, M., Do, R.K.G. and Togashi, K., 2018. Convolutional neural networks: an overview and application in radiology. *Insights Into Imaging*, 9(4), pp. 611-629.

Yao, H. et al., 2019. Dr2-net: Deep residual reconstruction network for image compressive sensing. *Neurocomputing*, 359, pp. 483-493.

Yu, W. K., and Liu, Y. M., 2019. Single-pixel imaging with origami pattern construction. *Sensors*, 19(23), pp. 5135.

Zhang, J. and Ghanem, B., 2018. ISTA-Net: Interpretable optimization-inspired deep network for image compressive sensing. s.l., *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 1828-1837.

Zhang, Z., Wang, X., Zheng, G. and Zhong, J., 2017. Hadamard single-pixel imaging versus Fourier single-pixel imaging. *Optics Express*, 25(16), pp. 19619-19639.

LIST OF PUBLICATIONS

1. Shin, Z. Y., Lin, H. S., Chai, T. Y., Wang, X., and Chua, S. Y., 2019. Programmable Single-Pixel Imaging. s.l., *Proceedings of the 13th International Conference on Sensing Technology (ICST)*, pp. 1-6.
2. Shin, Z., Lin, H.S., Chai, T.Y., Wang, X. and Chua, S.Y., 2021. Programmable spatially variant single-pixel imaging based on compressive sensing. *Journal of Electronic Imaging*, 30(2), pp. 021004.