SINGLE-IMAGE SUPER-RESOLUTION VIA
REGULARIZED EXTREME LEARNING REGRESSION FOR
IMAGERY
FROM MICROGRID POLARIMETERS

Thesis

Submitted to

The School of Engineering of the
UNIVERSITY OF DAYTON

In Partial Fulfillment of the Requirements for

The Degree of

Master of Science in Electrical Engineering

By

Garrett Craig Sargent

UNIVERSITY OF DAYTON

Dayton, Ohio

May, 2017

SINGLE-IMAGE SUPER-RESOLUTION VIA REGULARIZED EXTREME
LEARNING REGRESSION FOR IMAGERY FROM MICROGRID POLARIMETERS

Name: Sargent, Garrett Craig

APPROVED BY:

Vijayan K. Asari, Ph.D.
Advisor Committee Chairman
Professor, Department of Electrical and
Computer Engineering

Bradley M Ratliff, Ph.D.
Committee Member
Senior Image Processing Engineer,
UDRI Sensor APEX

Eric J. Balster, Ph.D.
Committee Member
Associate Professor, Department of
Electrical and Computer Engineering

Theus H. Aspiras, Ph.D.
Committee Member
Research Engineer, Department of
Electrical and Computer Engineering

Robert J. Wilkens, Ph.D., P.E.
Associate Dean for Research and
Innovation
Professor
School of Engineering

Eddy M. Rojas, Ph.D., M.A., P. E.
Dean, School of Engineering

© Copyright by
Garrett Craig Sargent
All rights reserved
2017

ABSTRACT

SINGLE-IMAGE SUPER-RESOLUTION VIA REGULARIZED EXTREME LEARNING REGRESSION FOR IMAGERY FROM MICROGRID POLARIMETERS

Name: Sargent, Garrett Craig
University of Dayton

Advisor: Dr. Vijayan K. Asari

Division of focal plane imaging polarimeters have the distinct advantage of being capable of obtaining temporally synchronized intensity measurements across a scene; however, they sacrifice spatial resolution in doing so due to their spatially modulated arrangement of the pixel-to-pixel polarizers and often result in aliased imagery. This shortcoming is often overcome through advanced demosaicing strategies that minimize the effects of false polarization while preserving as much high frequency content as possible. While these techniques can yield acceptable imagery, they tend to be computationally complex and the spatial resolution is often reduced below the native capabilities of the focal plane array. This thesis proposes a super-resolution method based upon a previously trained regularized extreme learning regression (RELR) that aims to recover missing high-frequency content beyond the spatial resolution of the sensor and correct low-frequency content, while maintaining good contrast between polarized and unpolarized artifacts presented in this thesis. For each of the four channels of the image, the modified RELR predicts the missing high-frequency and low-frequency components that result from undersampling and upsampling. These missing high-frequency components are then refined with a high pass filter and added back to the

upsampled image. This provides a fast and computationally simple way of recovering missing high frequency components that are lost with current state-of-the-art demosaicing algorithms. The modified RELR provides better results than other visible band single-image super-resolution techniques and is computationally fast, thus making it applicable to real-time applications. The obtained results demonstrate the effectiveness of the modified RELR for a truth scenario (no aliasing resulting from undersampling) and a derived microgrid scenario (aliasing resulting from undersampling). The truth scenario shows that the modified RELR performs exceptionally better than other algorithms, however, the derived microgrid scenario demonstrates the problems that result from aliasing for single-image super-resolution algorithms. In general, for the degree of linear polarization ($DoLP$) image product, aliasing greatly distorts objects within a scene and none of the super-resolution algorithms could do anything to correct for it. The modified RELR demonstrated superior performance against other super-resolution algorithms investigated at maintaining contrast between the polarized and unpolarized artifacts, which is of great importance. Future work is dedicated to coming up with fast ways to handle aliasing that is present in true microgrid imagery.

For my future wife Jennifer and my parents who have been supportive and encouraging every step
of my journey to pursue higher education.

ACKNOWLEDGMENTS

I would like to acknowledge the Jamestown Area Foundation for financially supporting me through my bachelor's degrees and masters' degree. The Bruce Higham scholarship awarded to me by the Jamestown Area Foundation was vital to my success here at the University of Dayton. I would also like to thank Dr. Brad Ratliff and Dr. Vijayan Asari for being my mentors and teaching me how to be a better researcher.

TABLE OF CONTENTS

ABSTRACT	iii
DEDICATION	v
ACKNOWLEDGMENTS	vi
LIST OF FIGURES	ix
LIST OF TABLES	xi
I. INTRODUCTION	1
II. LITERATURE SURVEY	9
2.1 Polarimetric Super-Resolution Techniques	9
2.2 Visible Band Single-Image Super-Resolution Techniques	9
2.3 Machine Learning Approach for Single-Image Super-Resolution	12
III. REGULARIZED EXTREME LEARNING REGRESSION FOR SUPER-RESOLUTION	14
3.1 Extreme Learning Machine	14
3.2 Extreme Learning Machine for Single-Image Super-Resolution	16
3.3 Regularized Extreme Learning Regression for Single-Image Super-Resolution . .	19
3.4 Modified Regularized Extreme Learning Regression for Polarimetric Imagery . .	24
IV. RESULTS AND ANALYSIS	31
4.1 Dataset	31
4.2 Testing Procedure	32
4.3 Truth Polarimetric Imagery	33
4.3.1 Results for $\times 2$ Magnification	33
4.3.2 Results for $\times 4$ Magnification	39
4.4 Derived Microgrid Polarimetric Imagery	43

4.4.1	Results for $\times 2$ Magnification	43
4.4.2	Results for $\times 4$ Magnification	49
4.5	Runtime Comparison	53
4.6	Discussion	55
V.	CONCLUSION	59
	BIBLIOGRAPHY	60

LIST OF FIGURES

1.1	Three images observed in Tyo et al. (a) RGB image with two trucks (b) IR image with two trucks (c) IR polarized image with two trucks	2
1.2	Focal plane layout of a microgrid	3
1.3	Derived microgrid raw image from a microgrid polarizer	3
1.4	Demosaicing Microgrid	4
1.5	Derived microgrid intensity channels	6
1.6	Truth intensity channels	7
3.1	ELM model	15
3.2	ELM training flow diagram	17
3.3	ELM testing flow diagram	18
3.4	RELR training flowchart	20
3.5	Robinson compass kernels	21
3.6	Alternative Robinson compass kernels	22
3.7	RELR testing flowchart	24
3.8	C vs SSIM, magnification factor of 2	26
3.9	C vs SSIM, magnification factor of 4	27
3.10	RELR training flowchart for HF prediction	28

3.11	RELR training flowchart for LF prediction	29
3.12	Modified RELR testing	30
4.1	Truth scenario, $\times 2$ magnification comparison for S_0	37
4.2	Truth scenario, $\times 2$ magnification comparison for $DoLP$	38
4.3	Truth scenario, $\times 4$ magnification comparison for S_0	42
4.4	Truth scenario, $\times 4$ magnification comparison for $DoLP$	43
4.5	Derived microgrid scenario, $\times 2$ magnification comparison for S_0	47
4.6	Derived microgrid scenario, $\times 2$ magnification comparison for $DoLP$	48
4.7	Derived microgrid scenario, $\times 4$ magnification comparison for S_0	52
4.8	Derived microgrid scenario, $\times 4$ magnification comparison for $DoLP$	53
4.9	Truth scenario, $\times 2$ magnification comparison for S_0	56
4.10	Derived microgrid scenario, $\times 2$ magnification comparison for S_0	57

LIST OF TABLES

4.1	SSIM S_0 , Truth Scenario, $\times 2$ Magnification	33
4.2	SSIM $DoLP$, Truth Scenario, $\times 2$ Magnification	34
4.3	PSNR S_0 , Truth Scenario, $\times 2$ Magnification	34
4.4	PSNR $DoLP$, Truth Scenario, $\times 2$ Magnification	35
4.5	RMSE S_0 , Truth Scenario, $\times 2$ Magnification	35
4.6	RMSE $DoLP$, Truth Scenario, $\times 2$ Magnification	36
4.7	SSIM S_0 , Truth Scenario, $\times 4$ Magnification	39
4.8	SSIM $DoLP$, Truth Scenario, $\times 4$ Magnification	39
4.9	PSNR S_0 , Truth Scenario, $\times 4$ Magnification	40
4.10	PSNR $DoLP$, Truth Scenario, $\times 4$ Magnification	40
4.11	RMSE S_0 , Truth Scenario, $\times 4$ Magnification	41
4.12	RMSE $DoLP$, Truth Scenario, $\times 4$ Magnification	41
4.13	SSIM S_0 , Derived Microgrid Scenario, $\times 2$ Magnification	44
4.14	SSIM $DoLP$, Derived Microgrid Scenario, $\times 2$ Magnification	44
4.15	PSNR S_0 , Derived Microgrid Scenario, $\times 2$ Magnification	45

4.16	PSNR $DoLP$, Derived Microgrid Scenario, $\times 2$ Magnification	45
4.17	RMSE S_0 , Derived Microgrid Scenario, $\times 2$ Magnification	46
4.18	RMSE $DoLP$, Derived Microgrid Scenario, $\times 2$ Magnification	46
4.19	SSIM S_0 , Derived Microgrid Scenario, $\times 4$ Magnification	49
4.20	SSIM $DoLP$, Derived Microgrid Scenario, $\times 4$ Magnification	49
4.21	PSNR S_0 , Derived Microgrid Scenario, $\times 4$ Magnification	50
4.22	PSNR $DoLP$, Derived Microgrid Scenario, $\times 4$ Magnification	50
4.23	RMSE S_0 , Derived Microgrid Scenario, $\times 4$ Magnification	51
4.24	RMSE $DoLP$, Derived Microgrid Scenario, $\times 4$ Magnification	51
4.25	Run time in seconds with modified RELR as reference , $\times 2$ magnification	54
4.26	Run time in seconds with RELR as reference, $\times 4$ magnification	55

CHAPTER I

INTRODUCTION

The physical quantities associated with optical fields are the intensity, wavelength, coherence, and polarization [1]. Typically, the intensity of optical radiation across a scene over a waveband of interest is collected. However, unlike polarization, the intensity of a scene does not describe the vector nature of the optical field. Polarization provides key information about the surface features, shape, shading, and roughness of objects within a scene. Another distinct advantage polarization provides is its inherent nature of being uncorrelated with spectral and intensity images [1]. This makes polarization useful for augmenting scenes from collected data that has no information about the vector nature of the objects throughout a scene. Figure 1.1, taken from [1], provides an insight into how polarization can provide enhanced contrast, where there is little to no contrast in the intensity image.

Figure 1.1: Three images observed in Tyo et al. (a) RGB image with two trucks (b) IR image with two trucks (c) IR polarized image with two trucks

Division of focal plane imaging polarimeters (also referred to as microgrid polarimeters) is one useful method for collecting polarization information from a scene. These microgrid polarimeters have the ability to obtain temporally synchronized intensity measurements across a scene. Each collected raw image has a 2×2 microgrid associated with each pixel, similar to the concept of the Bayer pattern used in color cameras [2]. Figure 1.2 demonstrates the layout of a microgrid associated with collected polarimetric imagery [2]. This microgrid uses four linear polarization filters of differing orientations (0° , 45° , 90° , and 135°) and will be used throughout the thesis. Figure 1.3 shows a derived microgrid raw image that would be obtained from a microgrid with the above mentioned polarizers. Figure 1.4, demonstrates demosaicing the microgrid to acquire the individual intensity channels.

Figure 1.2: Focal plane layout of a microgrid

Figure 1.3: Derived microgrid raw image from a microgrid polarizer

Figure 1.4: Demosaicing Microgrid

The goal of using a microgrid polarimeter is to estimate the Stokes vector

$$S = \begin{bmatrix} s_0 \\ s_1 \\ s_2 \\ s_3 \end{bmatrix}, \quad (1.1)$$

where s_0, s_1, s_2, s_3 are the symbols used to represent each element in the Stokes vector. The Stokes vector is a four-element real vector that describes polarized light [1] and can be used in order to determine the degree of polarization (DoP). The *DoP* is a common polarization image product and can be calculated using the Stokes vector elements by

$$DoP = \frac{\sqrt{s_1^2 + s_2^2 + s_3^2}}{s_0}. \quad (1.2)$$

The parameters of the Stokes vector described in Eq. 1.1 are defined as

$$s_0 = (I_0 + I_{45} + I_{90} + I_{135})/2 \quad (1.3)$$

$$s_1 = I_0 - I_{90} \quad (1.4)$$

$$s_2 = I_{45} - I_{135}, \quad (1.5)$$

where I_0 , I_{45} , I_{90} , and I_{135} refer to the intensity channels associated with 0° , 45° , 90° , and 135° polarizers, respectively. The Stokes parameter s_3 is the difference between right and left circularly polarized light but is often omitted due to the lack of observation in nature [2]. Thus, Eq. 1.2 becomes the degree of linear polarization (*DoLP*),

$$DoLP = \frac{\sqrt{s_1^2 + s_2^2}}{s_0}. \quad (1.6)$$

The goal of any demosaicing technique for microgrid imagery is to estimate the Stokes vector with as few artifacts as possible [2]. However, many problems arise when demodulation and interpolation is performed in order to reconstruct the Stokes vector. Since each pixel has a different instantaneous field of view (IFOV), when measurements are combined, the Stokes vector may contain errors across regions where there are edges [2]. In [2], an adaptive demosaicing strategy is developed in order to mitigate typical demosaicing artifacts that arise from optimal linear filtering reconstruction strategies, such as aliasing, zippering effects, and the loss of high frequency components. This adaptive demosaicing technique, while capable of preserving some HF content, is relatively slow but still useful for real-time applications. Thus, any super resolution techniques applied after the adaptive demosaicing method needs to be computationally fast. Figure 1.5 demonstrates the corresponding demosaiced channels with an adaptive demosaicing strategy developed in [2]. Figure 1.6 shows the same channels but from a rotating polarizer polarimeter where demosaicing is not a necessary step because each channel is collected separately. These intensity channels collected from the rotating polarizer polarimeter, are considered truth since they do not suffer from undersampling.

(a) Derived microgrid 0° intensity channel

(b) Derived microgrid 45° intensity channel

(c) Derived microgrid 90° intensity channel

(d) Derived microgrid 135° intensity channel

Figure 1.5: Derived microgrid intensity channels

(a) Truth 0° intensity channel

(b) Truth 45° intensity channel

(c) Truth 90° intensity channel

(d) Truth 135° intensity channel

Figure 1.6: Truth intensity channels

Many single-image and multi-image super resolution strategies exist for RGB data, however, very little have been applied to polarimetric imagery. The only such super-resolution technique applied to imagery from microgrid polarimeters is discussed in [3]. This technique is a multi-channel multi-frame super resolution algorithm that exploits correlation between the polarimetric channels. While this proposed technique is useful for addressing aliasing and other degradations,

it is computationally expensive and not practical for real-time applications. In this thesis, a single-image super-resolution technique based on an extreme learning machine (ELM), as proposed in [4], is applied in order to recover missing HF components that result from demosaicing a microgrid image.

The remainder of this thesis is organized as follows: Chapter II gives a detailed literature survey that deals with super-resolution for the visible band and polarimetric data. Chapter III describes the ELM, its application in super-resolution, and the proposed method for applying an ELM to polarimetric data. Chapter IV shows objective and subjective results from vigorous testing. Finally, Chapter V makes final remarks and summarizes the work done throughout the thesis.

CHAPTER II

LITERATURE SURVEY

2.1 Polarimetric Super-Resolution Techniques

In [3], the authors discuss the only known multi-channel, multi-frame super-resolution algorithms aimed specifically for the problem of image restoration in microgrid polarization imagers. The authors demonstrate the ability of their proposed algorithms to address aliasing and other degradations associated with microgrid imagers, without sacrificing field of view or optical resolution. Two distinct algorithms are presented with the first being designed to exploit correlation between the polarimetric channels based on a form of regularized least squares with an iterative solution, and the second doing the same but with a faster adaptive Wiener filter. These new super-resolution methods are demonstrated to be capable of enhancing polarimetric imagery when compared with their independent channel counterparts.

2.2 Visible Band Single-Image Super-Resolution Techniques

In [5], the authors discuss an image-zooming technique capable of producing a HR image from a LR image, referred to as Directional Cubic Convolution (DCC). Cubic Convolution (CC) is a typical

method for image interpolation, and DCC is a simple extension that can adapt to varying edge structures of images. DCC is useful for preserving sharp edges and details of images when interpolation is performed, while at the same time suppressing artifacts that occur with CC interpolation.

The authors in [6] develop a fast single-image super-resolution method that learns multiple adaptive interpolation kernels, referred to as Multiple Kernel Learning (MKL). Each HR image patch is assumed to be sparsely represented by several simple image structures. These image structures are then assumed capable of being assigned a suitable interpolation kernel. MKL consists of clustering the training image patches into several classes and training each specified class' interpolation kernel. Then, for each LR image patch, a few suitable kernels are selected to make the final interpolation kernel. The authors also discuss how the efficiency of MKL can be guaranteed since it is mainly based on simple linear algebra computations.

The authors in [7] discuss a New Edge-Directed Interpolation (NEDI) algorithm for natural images. NEDI works by first estimating local covariance coefficients from LR imagery and then uses these estimates to interpolate at a higher resolution based on geometric duality between LR covariance and HR covariance. NEDI is a hybrid approach of switching between bilinear and covariance-based adaptive interpolation in order to reduce overall complexity. Results obtained are demonstrated to substantially improve on resolution enhancement of grayscale images compared to conventional linear interpolation methods.

The authors in [8] develop a single-image super-resolution method referred to as Iterative Curvature Based Interpolation (ICBI). ICBI is based on a two-step grid filling and an iterative correction of the interpolated pixels. These interpolated pixels are obtained by minimizing an objective function that is dependent on the second-order directional derivatives of the intensity associated with an image. ICBI is demonstrated to perform better, objectively and subjectively, when compared to NEDI with the addition of having a speed improvement of a one or two orders of magnitude.

In [9], the authors use Sparse Mixing Estimators (SME) for single-image super-resolution. This SME is a class of nonlinear inverse estimators obtained with adaptive mixing of linear estimators. This class of mixing estimators is then used for applications to image interpolation. The authors show how nonlinear algorithms account for adaptive image models that are often better than linear interpolators, which allows them to be considered super-resolution algorithms.

In [10], the authors use a Statistical Prediction Model (SPM) to address single-image super-resolution. This SPM is based on the sparse representations of LR and HR image patches. The SPM allows for the avoidance of any invariance assumption that is common among other sparsity-based approaches. The authors describe how HR image patches are obtained using a minimum mean square error estimation and has the useful interpretation of a feedforward neural network. The authors also discuss how further enhanced performance can be achieved by data clustering and cascading several levels of the algorithm. SPM is known for offering a desirable compromise between low computational complexity and reconstruction quality, when compared to other state-of-the-art super-resolution methods.

In [11], the authors introduce Anchored Neighborhood Regression (ANR), which is a fast super-resolution algorithm that does not compromise on speed. The authors show that the use of sparse learned dictionaries in combination with neighbor embedding methods can be used rather than just dictionaries. The authors also show how using global collaborative coding increases speed considerably by reducing the super-resolution mapping to a precomputed projective matrix. ANR is demonstrated to obtain similar or improved quality with an improvement in speed by one or two orders of magnitude.

2.3 Machine Learning Approach for Single-Image Super-Resolution

In [12], a single-image super-resolution technique, referred to as Support Vector Regression, is shown to have some super-resolving capabilities on dissimilar images. The main idea of SVR is to develop a function that maps a low-resolution (LR) patch and a central pixel of a high-resolution (HR) image patch. Results obtained from the developed SVR show that some high-frequency (HF) components are able to be recovered, whereas, peak signal-to-noise ratio (PSNR) improvement was not impressive [12]. The authors conjecture that better results are thought to be obtainable with larger training data bases.

In [13], a regression-based method for single-image super-resolution is introduced. This is accomplished via a Kernel Ridge Regression (KRR), which estimates HF details of an underlying HR image. The KRR was trained on 200,000 data points, which takes around a day to complete. However, processing time is much faster than SVR while offering competitive results.

In [14], a self-learning based approach to super-resolution. The authors advance SVR with image sparse representation that offers generalization in modeling the relationship between images and their associated super-resolution versions. This SVR method is different than most self-learning methods, as it does not assume the self-similarity of image patches within an image across different image scales, and instead, uses Bayes decision theory to select the optimal SVR model when reconstructing super-resolved images. However, since the SVR is trained with LIBSVM, only linear SVR models are considered.

In [15], the authors propose a single-image super-resolution technique based on an Extreme Learning Machine (ELM). The overall goal is to reconstruct HF components that contain details and fine structures that are typically missing from LR images. This is done by applying set of input features from LR images and using a set of HR images as a target. A model can then be obtained by

solving for a set of weights using least squares analysis that aims at minimizing error between the input features and the target output. The obtained model can then be used to predict HF components of LR input images that is then added back to interpolated LR image.

The above mentioned methods each have their own benefits in terms of performance based on speed, qualitative effectiveness, efficiency, etc. However, there is still a need for faster, more efficient, and better performing single-image super-resolution algorithms for real-time applications. In this thesis, a modified regularized extreme learning regression model is proposed that provides exceptional performance with real-time capabilities.

CHAPTER III

REGULARIZED EXTREME LEARNING REGRESSION FOR SUPER-RESOLUTION

Extreme learning machines are useful when speed and a generalized solution is a requirement. The accuracy and speed of the RELR model used in [4] provides great opportunity for recovering otherwise lost HF components resulting from demosaicing microgrid imagery. Multi-channel multi-frame super-resolution techniques, as discussed in [3], are the only such methods attempting to recover lost HF components in microgrid imagery by exploiting correlation in the polarimetric channels. However, approaches like this are computationally expensive and may not be realistic for real-time applications, especially considering the current state-of-the-art demosaicing techniques are computationally slow as well. In this thesis an advanced machine learning based approach for single-image super-resolution is proposed that provides real time capability with enhanced HF components. The systematic development of the new regularized extreme learning machine based approach is presented in this chapter.

3.1 Extreme Learning Machine

ELMs are a generalized single-hidden-layer feed-forward networks (SLFNs). In [16], the authors show the following four points: ELMs provide a unified learning platform that can be applied

in regression applications, ELMs have milder optimization constraints compared to other algorithms, other algorithms require higher complexity and achieve suboptimal performance, and ELMs can approximate any target continuous function. For the purposes of this thesis, a unified learning platform that can be applied in regression applications and refrain from complexity is presented. A standard ELM model can be viewed in Fig. 3.1, where w are the weights connecting input layer to hidden layer, b are the biases, h is any nonlinear activation function, β is the output weights between hidden layer and output nodes, L is the number of hidden nodes, d is the number of input neurons, and m is the number of output neurons.

Figure 3.1: ELM model

In an ELM, the hidden layer need not to be iteratively tuned. The output for a given node is given by

$$y_L(x) = \sum_{i=1}^L \beta_i h_i(x) = \mathbf{h}\beta, \quad (3.1)$$

where β is the output weight vector between the hidden layer of L nodes and the desired output node, in the form $\beta = [\beta_1, \beta_2, \dots, \beta_L]$. The output between the input and the hidden layer, $\mathbf{h}(x) =$

$[h_1(x), h_2(x), \dots, h_L(x)]$, maps the input from d-dimensional space to L-dimensional feature space.

This process is known as ELM feature mapping.

According to the authors in [16, 17, 18, 19], in ELM learning theory, an ELM can approximate any continuous target function $f(x)$.

$$\lim_{L \rightarrow +\infty} \|y_L(x) - f(x)\| = \lim_{L \rightarrow +\infty} \|\beta_i h_i(x) - f(x)\| = 0, \quad (3.2)$$

The goal of the ELM is to minimize the training error as well as the norm of the output weights by

$$\text{Minimize : } \|\mathbf{H}\boldsymbol{\beta} - \mathbf{T}\|^2 \quad \text{and} \quad \|\boldsymbol{\beta}\|, \quad (3.3)$$

where $\mathbf{H}$ is the hidden layer output matrix. Here $\boldsymbol{\beta}$ can be solved by least squares analysis

$$\hat{\boldsymbol{\beta}} = \mathbf{H}^\dagger \mathbf{T}, \quad (3.4)$$

where $\mathbf{H}^\dagger$ is the Moore Penrose generalized inverse of matrix $\mathbf{H}$, and is computed by $\mathbf{H}^\dagger = (\mathbf{H}^T \mathbf{H})^{-1} \mathbf{H}^T$. Different methods exist, depending on input size, on how to solve for $\boldsymbol{\beta}$. For instance, if the data size, N , is much larger than the number of hidden nodes, L , then a regularization term can be introduced. Depending upon application, a number of different activation functions can be used for $\mathbf{H}$, such as sigmoid, hard-limit, Gaussian, etc [16].

3.2 Extreme Learning Machine for Single-Image Super-Resolution

In Chapter II, a super-resolution method was discussed that utilizes an ELM. In [15], the authors describe how the generalization and simplicity of the ELM is a good match for predicting lost HF components from LR imagery. Here, the authors define super-resolution as the process of generating HR images from LR inputs, and describe how an ELM could be trained by example to recover HF content that contain details and fine structures that are not present in LR images. The training and testing flow diagram is shown in Figs. 3.2 and 3.3, respectively.

Figure 3.2: ELM training flow diagram

Figure 3.3: ELM testing flow diagram

Training begins by assigning HF components from HR images as target values. These HF components are computed by

$$I_{HF} = I_{HR} - I_0, \quad (3.5)$$

where I_{HR} is the original HR image, I_0 is the blurred, downsampled, and interpolated LR image. The image containing the HF components is labeled as I_{HF} . Image features from LR images are then fed into an ELM, as shown in 3.1, as inputs and a model is learned. The features consist of pixel intensity values from local image patches as well as 1st and 2nd order derivative magnitudes as described in Eq. 3.6. The total feature vector is thus $m^2 + 5$ in length, where m^2 is dependent on the $m \times m$ image patch. The 1st and 2nd order derivative filters, d_f , are described by

$$d_f = \left(\frac{\partial I_0}{\partial x}, \quad \frac{\partial I_0}{\partial y}, \quad \frac{\partial^2 I_0}{\partial x^2}, \quad \frac{\partial^2 I_0}{\partial y^2}, \quad \frac{\partial^2 I_0}{\partial x \partial y} \right). \quad (3.6)$$

For testing, this model is then used to predict HF content from LR test images and the results are added to the initially interpolated I_0 image.

3.3 Regularized Extreme Learning Regression for Single-Image Super-Resolution

In [4], the authors train an ELM to learn how to correct for lost HF components in low resolution (LR) RGB imagery. The RELR method contains two steps: (1) learning HF features from a set of high resolution (HR) images; and (2) predicting HF components from LR testing images. These steps are applied to RGB data sets. The training flow diagram can be observed in Fig. 3.4.

Figure 3.4: RELR training flowchart

Training sets are produced by blurring, downsampling, and then upsampling HR images. These training images, I_{LR} , are then subtracted from the original HR images, I_{HR} , in order to determine the lost HF components. To account for fine details of edge information, a derivative filter is applied

to a $p \times p$ local image patch at each pixel location in I_{LR} . The derivative filter chosen for this algorithm is the Robinson compass operator, which includes a single kernel mask that is rotated in eight directions in 45 degree increments. These eight kernels can be observed in Fig. 3.5.

$$\begin{aligned}
k_1 &= \begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix} & k_2 &= \begin{bmatrix} 0 & 1 & 2 \\ -1 & 0 & 1 \\ -2 & -1 & 0 \end{bmatrix} & k_3 &= \begin{bmatrix} 1 & 2 & 1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{bmatrix} & k_4 &= \begin{bmatrix} 2 & 1 & 0 \\ 1 & 0 & -1 \\ 0 & -1 & -2 \end{bmatrix} \\
k_5 &= \begin{bmatrix} 1 & 0 & -1 \\ 2 & 0 & -2 \\ 1 & 0 & -1 \end{bmatrix} & k_6 &= \begin{bmatrix} 0 & -1 & -2 \\ 1 & 0 & -1 \\ 2 & 1 & 0 \end{bmatrix} & k_7 &= \begin{bmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ 1 & 2 & 1 \end{bmatrix} & k_8 &= \begin{bmatrix} -2 & -1 & 0 \\ -1 & 0 & 1 \\ 0 & 1 & 2 \end{bmatrix}
\end{aligned}$$

Figure 3.5: Robinson compass kernels

Application of the Robinson compass kernels produces eight filtered images, which are concatenated with the local neighborhood of each pixel to form a feature vector:

$$f_z(s) = [P, f_1(s), f_2(s), \dots, f_8(s)], \quad (3.7)$$

where $z = [1, 2, \dots, 8]$ and P is the local neighborhood of each pixel. In [4], the authors discuss how the Robinson compass kernels can be replaced by eight corresponding filters as shown in Fig. 3.6, where a dot product instead of a convolution can be performed.

$$\begin{aligned}
l_1 &= \begin{bmatrix} 3 & 0 & -3 \\ 4 & 0 & -4 \\ 3 & 0 & -3 \end{bmatrix} \quad l_2 = \begin{bmatrix} 0 & -3 & -4 \\ 3 & 0 & -3 \\ 4 & 3 & 0 \end{bmatrix} \quad l_3 = \begin{bmatrix} -3 & -4 & -3 \\ 0 & 0 & 0 \\ 3 & 4 & 3 \end{bmatrix} \quad l_4 = \begin{bmatrix} -4 & -3 & 0 \\ -3 & 0 & 3 \\ 0 & 3 & 4 \end{bmatrix} \\
l_5 &= \begin{bmatrix} -3 & 0 & 3 \\ -4 & 0 & 4 \\ -3 & 0 & 3 \end{bmatrix} \quad l_6 = \begin{bmatrix} 0 & 3 & 4 \\ -3 & 0 & 3 \\ -4 & -3 & 0 \end{bmatrix} \quad l_7 = \begin{bmatrix} 3 & 4 & 3 \\ 0 & 0 & 0 \\ -3 & -4 & -3 \end{bmatrix} \quad l_8 = \begin{bmatrix} 4 & 3 & 0 \\ 3 & 0 & -3 \\ 0 & -3 & -4 \end{bmatrix}
\end{aligned}$$

Figure 3.6: Alternative Robinson compass kernels

For a given set of training points that are represented as $\{x_i, y_i\}_{i=1}^N$, where $x_i, y_i \in \mathbb{R}^d$ and N is the total number of training patches in an image, the following cost function is minimized for the regressor $f_z(x_i)$ as shown in [4]:

$$\epsilon(f_L) = \|f_z(x_i) - y_i\| = \left\| \sum_{j=1}^L \beta_j h_j(x_i) - y_i \right\|, \quad (3.8)$$

where β_j denotes the weights linking the j^{th} hidden node and the output nodes, $h_j(x_i) \in \mathbb{R}^L$ is the output of the j^{th} hidden-layer for an input x_i . The hidden-layer function $h_j(x_i)$ can be computed by any nonlinear piecewise continuous function. For the purpose of this thesis, a sigmoid function is chosen, thus $h_j(x_i)$ can be given by:

$$h_j(x_i) = G(w_j, b_j, x_i) = \frac{1}{1 + e^{-(w_j^T x_i + b_j)}}, \quad (3.9)$$

where $w_j \in \mathbb{R}^d$ represents the weights connecting the j^{th} hidden node to the input nodes, and b_j is the biases for each of the corresponding weights. Both, w_j and b_j are randomly assigned prior to the training data. The overall goal is to produce the HF components of $\mathbf{I}_{HR}$ given the input, x_i , feature vector described in Eq. 3.7:

$$\sum_{j=1}^L \beta_j h_j(x_i) = y_i. \quad (3.10)$$

For N number of input data, Eq. 3.10 can be represented in matrix notation as:

$$\mathbf{H}\boldsymbol{\beta} = \mathbf{Y}, \quad (3.11)$$

where $\mathbf{Y}$ is the desired output, $\boldsymbol{\beta} \in \mathbb{R}^{L \times m}$, and $\mathbf{H}$ is a matrix representing $h_j(x_i)$. $\boldsymbol{\beta}$ is also referred to as the output weight matrix and what is to be solved for moving forward. In [16], extensive research is done as to how to best solve for $\boldsymbol{\beta}$ given different number of input sizes. In general, solving for $\boldsymbol{\beta}$ is done by ordinary least-square error analysis [16]:

$$\hat{\boldsymbol{\beta}} = \mathbf{H}^\dagger \mathbf{Y}, \quad (3.12)$$

Moreover, both [16, 20] apply a regularization term in order to avoid overfitting due to the large nature of the input data:

$$\mathbf{P}(x_i) = h(x_i) \boldsymbol{\beta} = h(x_i) \left(\frac{\mathbf{I}}{C} + \mathbf{H}\mathbf{H}^T \right)^{-1} \mathbf{H}^T \mathbf{Y}, \quad (3.13)$$

where $\mathbf{P}(x_i)$ is the predicted HF components, $\mathbf{I}$ is the identity matrix, and C is a user specified parameter that provides a tradeoff between the distance of the separating margin and error in training.

In order to alleviate undesired low frequency components, a high-pass (HP) filter is applied to the predicted output $\mathbf{P}(x_i)$ before it is added to the bicubically interpolated $\mathbf{I}_{LR}$. Experimental results from [4] show the HP filter improves overall accuracy while not affecting the speed performance of the algorithm. The final output of the RELR algorithm is obtained by adding in missing HF components to its corresponding bicubically interpolated image. Thus, the final output can be computed as:

$$\mathbf{E}_{HR} = F_{HP}^{v \times v}(\mathbf{P}) + \mathbf{I}_{LR}, \quad (3.14)$$

where $F_{HP}^{v \times v}(\cdot)$ represents a HP filter of size $v \times v$, and $\mathbf{P}$ is the predicted HR image. The testing flow diagram can be viewed in Fig. 3.7.

Figure 3.7: RELX testing flowchart

3.4 Modified Regularized Extreme Learning Regression for Polarimetric Imagery

Typically, microgrid polarimeters collect intensity channels at four orientations (0° , 45° , 90° , and 135°). These images suffer from the undersampling when demosaicing is performed. This

results in aliasing and a loss of HF components. It is envisaged that preserving LF contents will help the performance of recovering HF content, while reducing the amount of additional noise.

A modification of the RELR is suggested in order to accommodate polarimetric imagery. This modified RELR consists of applying the RELR discussed in [4], with an additional RELR that learns a model based on a multi-scale Gaussian input. Starting with the RELR discussed in [4], normalization constraints are removed in order to account for difficulties that come with normalizing data acquired from microgrid acquisition systems. Typically, raw data acquired from microgrid acquisition systems are fed into a calibration procedure, followed by a process to acquire a desired image product, and finally scaled to fit a display on a screen. Considering that super-resolution is being applied in the steps between calibration and acquiring an image product, rather than developing a reverse normalization procedure to get back the true intensity information, normalization is bypassed all together.

When solving for the output weights using Eq. 3.13, an optimized value for C is considered. According to [16], the value of C in Eq. 3.13 is in the set $\{2^{-24}, 2^{-23}, \dots, 2^{24}, 2^{25}\}$. In order to find the optimized parameter value for C , a number of simulations are run in order to check for what values of C give the best structural similarity (SSIM) [21] for a given number of hidden nodes. Figures 3.8 and 3.9 show the relationship between a subset of the values of C and SSIM for magnification factors of 2 and 4, respectively. Based on the results obtained, for this particular data set, a value of C between 2^{10} and 2^{15} gives the best performance for hidden nodes in the range of 70 to 90. Given that computation time increases with the number of nodes for computations performed on a CPU, the range of 70 to 90 hidden nodes is chosen due to the lack a of significant increase in performance with more hidden nodes.

Figure 3.8: C vs SSIM, magnification factor of 2

Figure 3.9: C vs SSIM, magnification factor of 4

The flow diagram for HF prediction using the optimized RELR is shown in Fig. 3.10.

Figure 3.10: RELR training flowchart for HF prediction

Next, in order to obtain better performance, a modified RELR is proposed in order to account for the LF components associated with an image. A multi-scale Gaussian feature vector is used for an LF RELR model with LF components of the desired HR image as target values. The LR input image is filtered with three Gaussian filters with different standard deviations in order to generate a feature vector. These 2-D Gaussian kernels are determined by Eqs. 3.15 and 3.16, as

$$h_g(i, j) = e^{-\left(\frac{i^2 + j^2}{2\sigma^2}\right)}, \quad (3.15)$$

and

$$h(i, j) = \frac{h_g(i, j)}{\sum_{i=1}^N \sum_{j=1}^N h_g(i, j)}, \quad (3.16)$$

where $h(i, j)$ is the final output kernel given a standard deviation, σ , and i and j are the two dimensional indices. Standard deviations of 1, 2, and 3 were chosen for the three Gaussian filters.

The output of the LF RELR is then used as the input image of the HF RELR optimized earlier. This provides for an increase in performance with a minimal reduction in speed. The LF RELR architecture is shown in Fig. 3.11, and the entire modified RELR testing process is shown in Fig. 3.12.

Figure 3.11: RELR training flowchart for LF prediction

Figure 3.12: Modified RELR testing

CHAPTER IV

RESULTS AND ANALYSIS

Experiments were conducted to demonstrate the capability of single-image super-resolution techniques in polarimetric imagery. The performance of the algorithms were tested on a set of truth data and a derived microgrid polarimetric imagery. The effectiveness of the algorithms were tested for magnification factors of 2 and 4. Subjective and objective evaluations were performed and the results are presented in this chapter.

4.1 Dataset

For the visible waveband, 14 images are taken from a Nikon D750 camera with an attached polarizer that was used to collect intensity channels at four orientations (0° , 45° , 90° , and 135°). This brings the total number of images up to 56 (4 intensity channels per image). These images are considered to be the HR images that are used for training. Since the images that are collected are not from a microgrid and do not suffer undersampling, they can be used as a truth data set for demosaicing a microgrid polarimeter. For testing purposes, 10 microgrid images are derived from the set of HR images and the adaptive demosaicing algorithm discussed in [2], is used to demosaic the channels in order to simulate the microgrid Stokes estimation process. This results in a total of 40 LR images for testing purposes. The original HR images are also used for testing in order to compare truth performance against performance from derived microgrid imagery.

4.2 Testing Procedure

The modified RELR and RELR were vigorously tested on a set of 10 testing images for both the truth and derived microgrid case. There are 4 channels associated with each testing image, therefore, a total of 40 images were used. The products discussed in Eqs. 1.3 and 1.6 are used for the objective and subjective comparisons of the various single-image super-resolution techniques at resolution factors of 2 and 4.

Seven of the various methods that are discussed in [4] are used for comparisons: directional cubic convolution (DCC) [5]; multiple kernel learning (MKL) [6]; new edge-directed interpolation (NEDI) [7]; iterative curvature based interpolation (ICBI) [8]; sparse mixing estimators (SME) [9]; statistical prediction model (SPM) [10]; and anchored neighborhood regression (ANR) [11]. These seven methods were chosen based on their speed and performance and are modified to account for the added stipulation that normalization is not performed on the input imagery. Implementation of all the algorithms were found on the respective authors' websites. In addition to these seven super-resolution algorithms, an unsharp mask (UM), bilinear interpolation (BLI), and bicubic interpolation (BCI) are also used for comparison. Default parameters and settings are used throughout the testing process.

Each super resolved image is quantitatively compared with the original HR image in order to determine the effectiveness of each single-image super-resolution algorithm. These quantitative evaluation methods include structural similarity method (SSIM) [21], peak signal-to-noise ratio (PSNR), and root mean square error (RMSE). SSIM produces an index between 0 and 1 based on a quality assessment based on luminance, contrast, and structure, whereas PSNR compares image quality with respect to noise. The RMSE is simply calculated as another form of quantitative measurement.

A speed performance is also determined for each of the algorithms and is of great value for real-time applications. The tables below provide detailed results of all the performance evaluations. In the tables, the best performing algorithm is highlighted in green, whereas, the second best is highlighted in yellow. The figures demonstrate how each algorithm performs visually and is useful for subjective evaluation.

4.3 Truth Polarimetric Imagery

4.3.1 Results for $\times 2$ Magnification

Tables 4.1 and 4.2 demonstrate the effectiveness of the modified RELR for the S_0 and $DoLP$ polarimetric image products when SSIM is of importance. The modified RELR outperforms all other techniques for both image products. The second best algorithm in both cases is the RELR, with MKL following in third. These are the only three super-resolution algorithms that outperform an unsharp mask.

Table 4.1: SSIM S_0 , Truth Scenario, $\times 2$ Magnification

SR Algorithm	Image 1	Image 2	Image 3	Image 4	Image 5	Image 6	Image 7	Image 8	Image 9	Image 10	Average
BLI	0.67537	0.69648	0.81576	0.68975	0.68154	0.70463	0.63372	0.71820	0.55375	0.69243	0.68616
BCI	0.70616	0.72546	0.83966	0.72345	0.71032	0.73427	0.67911	0.76168	0.59247	0.71977	0.71924
UM	0.72267	0.74099	0.85211	0.74136	0.72564	0.75002	0.70308	0.78388	0.61394	0.73398	0.73677
MKL	0.72834	0.74591	0.85601	0.74740	0.73063	0.75533	0.71129	0.79084	0.62062	0.74003	0.74264
DCC	0.68363	0.70486	0.81894	0.69935	0.68901	0.71165	0.65277	0.73107	0.56705	0.69900	0.69573
SPM	0.71242	0.73125	0.84461	0.73026	0.71629	0.74039	0.68770	0.76936	0.60040	0.72547	0.72582
NEDI	0.67982	0.70132	0.81834	0.69449	0.68540	0.70772	0.64219	0.72244	0.56008	0.69520	0.69070
ICBI	0.68634	0.70794	0.82397	0.70204	0.69182	0.71440	0.65201	0.73023	0.56845	0.70127	0.69785
SME	0.69573	0.71641	0.83167	0.71176	0.70080	0.72320	0.66475	0.74310	0.57823	0.71011	0.70758
ANR	0.71554	0.73417	0.84574	0.73455	0.71909	0.74367	0.69457	0.77393	0.60651	0.72813	0.72959
RELR	0.75549	0.77212	0.87596	0.77751	0.75630	0.77960	0.75001	0.82249	0.65863	0.76087	0.77090
Modified RELR	0.76633	0.78288	0.88221	0.78960	0.76611	0.78966	0.76513	0.83198	0.67619	0.76987	0.78200

Table 4.2: SSIM $DoLP$, Truth Scenario, $\times 2$ Magnification

SR Algorithm	Image 1	Image 2	Image 3	Image 4	Image 5	Image 6	Image 7	Image 8	Image 9	Image 10	Average
BLI	0.56451	0.54630	0.61192	0.50431	0.44413	0.30861	0.39067	0.24140	0.29011	0.60784	0.45098
BCI	0.58208	0.55824	0.63154	0.52553	0.46463	0.32471	0.42139	0.28454	0.31414	0.62620	0.47330
UM	0.59201	0.56529	0.64377	0.53817	0.47636	0.33403	0.43945	0.31020	0.32816	0.63529	0.48627
MKL	0.59705	0.56997	0.64847	0.54472	0.48282	0.34243	0.44804	0.32560	0.33536	0.64298	0.49374
DCC	0.56581	0.55143	0.61909	0.51219	0.44877	0.31587	0.40649	0.26673	0.29888	0.61284	0.45981
SPM	0.58591	0.56090	0.63562	0.53013	0.46931	0.32846	0.42786	0.29361	0.31930	0.63013	0.47812
NEDI	0.56765	0.54981	0.61579	0.50867	0.44852	0.31217	0.39916	0.25305	0.29440	0.61139	0.45606
ICBI	0.56976	0.55164	0.61862	0.51162	0.45132	0.31460	0.40302	0.25931	0.29813	0.61341	0.45914
SME	0.57544	0.55509	0.62437	0.51737	0.45774	0.31976	0.41184	0.27176	0.30425	0.61972	0.46573
ANR	0.58482	0.56286	0.63956	0.53416	0.46981	0.33332	0.43443	0.30663	0.32414	0.63117	0.48209
RELR	0.60710	0.57665	0.66185	0.55818	0.49533	0.34904	0.47005	0.34978	0.35283	0.64614	0.50670
Modified RELR	0.61334	0.58381	0.67157	0.57127	0.50441	0.35789	0.48525	0.36449	0.36461	0.65331	0.51700

Comparing PSNR of the testing images, tables 4.3 and 4.4 indicate that the modified RELR is performing best on the chosen set of testing images, followed closely by RELR, and closely again by MKL for the S_0 image product. For the $DoLP$ image product, the modified RELR performs the best and an the RELR performs second best.

Table 4.3: PSNR S_0 , Truth Scenario, $\times 2$ Magnification

SR Algorithm	Image 1	Image 2	Image 3	Image 4	Image 5	Image 6	Image 7	Image 8	Image 9	Image 10	Average
BLI	27.32097	26.13484	28.32507	25.11668	26.70602	26.88896	23.62312	25.82726	24.50102	25.76922	26.02132
BCI	27.85549	26.62624	28.87982	25.64406	27.28690	27.56893	24.34970	26.70992	24.98281	26.45881	26.63627
UM	28.12256	26.85777	29.11807	25.88958	27.57933	27.91824	24.73733	27.19305	25.22977	26.81906	26.94648
MKL	28.21706	26.94441	29.18088	25.97388	27.68098	28.03224	24.86802	27.33887	25.32054	26.98356	27.05404
DCC	24.29105	22.35399	22.46799	20.86433	23.53098	23.66350	21.33458	24.25290	22.19043	23.05559	22.80053
SPM	27.96942	26.73734	29.01888	25.76412	27.41302	27.70304	24.48916	26.86130	25.08201	26.59476	26.76331
NEDI	27.20191	26.01892	28.08292	24.95000	26.57748	26.72689	23.68363	25.64259	24.45311	25.62049	25.89579
ICBI	27.40357	26.26403	28.47603	25.22262	26.78632	26.94740	23.90688	25.77954	24.61364	25.81011	26.12101
SME	27.61705	26.47496	28.75898	25.45034	27.02947	27.20603	24.16997	26.05569	24.77725	26.06801	26.36077
ANR	25.14905	23.12294	23.66053	21.80259	24.50335	24.84536	22.20234	25.56063	23.05439	24.21480	23.81160
RELR	28.60119	27.24768	29.58742	26.36315	28.18480	28.60585	25.48444	28.03298	25.67640	27.63482	27.54187
Modified RELR	28.80501	27.49219	29.97760	26.66654	28.41003	28.83399	25.81664	28.21448	25.86701	27.87937	27.79629

Table 4.4: PSNR $DoLP$, Truth Scenario, $\times 2$ Magnification

SR Algorithm	Image 1	Image 2	Image 3	Image 4	Image 5	Image 6	Image 7	Image 8	Image 9	Image 10	Average
BLI	23.21162	22.31976	17.41440	16.61798	18.94914	14.33103	15.61176	12.60600	14.90593	19.81620	17.57838
BCI	23.40201	22.58251	17.86300	17.06607	19.20237	14.63229	15.97014	13.15700	15.36129	20.25554	17.94922
UM	23.49772	22.74676	18.15145	17.34935	19.34816	14.82162	16.19263	13.50398	15.63594	20.47399	18.17216
MKL	23.56517	22.79023	18.21094	17.38859	19.42600	14.88478	16.24831	13.59407	15.68355	20.60481	18.23964
DCC	22.63730	22.49411	17.67548	16.92396	18.87501	14.51534	15.77649	12.97038	15.07832	20.06905	17.70154
SPM	23.44203	22.62771	17.93916	17.13758	19.25498	14.68826	16.03337	13.25251	15.43520	20.32778	18.01386
NEDI	23.24585	22.41328	17.56730	16.76620	18.98016	14.42780	15.68731	12.80088	14.99944	19.94190	17.68301
ICBI	23.25775	22.47490	17.64372	16.85578	19.01005	14.48607	15.74673	12.90558	15.08261	20.00422	17.74674
SME	23.32439	22.53715	17.75433	16.97170	19.08359	14.57308	15.83501	13.06256	15.19708	20.15479	17.84937
ANR	22.92013	22.68264	18.00773	17.23328	19.15703	14.74901	16.09720	13.35856	15.47427	20.37311	18.00530
RELNR	23.60196	23.04892	18.69797	17.80857	19.57295	15.20303	16.55689	14.09965	16.09436	20.69962	18.53839
Modified RELNR	23.62343	23.18654	18.91167	18.08158	19.65879	15.30672	16.73000	14.26584	16.21469	20.79407	18.67733

Finally, Tables 4.5 and 4.6 demonstrate that for the S_0 image product, the modified RELNR and RELNR provide the least amount of error per pixel on average, with the modified RELNR being the best and the RELNR being second best. As expected, due to the PSNR performance, the modified RELNR provides the least error per pixel and the RELNR provides the second least for the $DoLP$ image product.

Table 4.5: RMSE S_0 , Truth Scenario, $\times 2$ Magnification

SR Algorithm	Image 1	Image 2	Image 3	Image 4	Image 5	Image 6	Image 7	Image 8	Image 9	Image 10	Average
BLI	0.04305	0.04935	0.03835	0.05548	0.04621	0.04524	0.06589	0.05113	0.05956	0.05147	0.05057
BCI	0.04048	0.04663	0.03598	0.05222	0.04322	0.04184	0.06061	0.04619	0.05635	0.04754	0.04710
UM	0.03925	0.04541	0.03500	0.05076	0.04179	0.04019	0.05796	0.04369	0.05477	0.04561	0.04544
MKL	0.03883	0.04496	0.03475	0.05027	0.04130	0.03966	0.05710	0.04296	0.05420	0.04475	0.04488
DCC	0.06102	0.07626	0.07527	0.09053	0.06660	0.06559	0.08576	0.06129	0.07771	0.07034	0.07304
SPM	0.03995	0.04604	0.03540	0.05150	0.04259	0.04120	0.05964	0.04539	0.05571	0.04680	0.04642
NEDI	0.04364	0.05001	0.03943	0.05656	0.04689	0.04610	0.06544	0.05222	0.05989	0.05236	0.05125
ICBI	0.04264	0.04862	0.03769	0.05481	0.04578	0.04494	0.06378	0.05141	0.05879	0.05123	0.04997
SME	0.04161	0.04745	0.03648	0.05339	0.04452	0.04362	0.06187	0.04980	0.05769	0.04973	0.04862
ANR	0.05528	0.06980	0.06561	0.08126	0.05954	0.05724	0.07760	0.05272	0.07035	0.06155	0.06510
RELNR	0.03715	0.04341	0.03316	0.04807	0.03897	0.03713	0.05318	0.03966	0.05202	0.04152	0.04243
Modified RELNR	0.03629	0.04221	0.03170	0.04642	0.03798	0.03617	0.05119	0.03884	0.05089	0.04037	0.04120

Table 4.6: RMSE $DoLP$, Truth Scenario, $\times 2$ Magnification

SR Algorithm	Image 1	Image 2	Image 3	Image 4	Image 5	Image 6	Image 7	Image 8	Image 9	Image 10	Average
BLI	0.06909	0.07656	0.13467	0.14761	0.11286	0.19207	0.16573	0.23426	0.17976	0.10214	0.14148
BCI	0.06759	0.07428	0.12789	0.14018	0.10962	0.18552	0.15904	0.21986	0.17058	0.09710	0.13517
UM	0.06685	0.07289	0.12372	0.13569	0.10779	0.18152	0.15501	0.21125	0.16527	0.09469	0.13147
MKL	0.06633	0.07253	0.12287	0.13507	0.10683	0.18020	0.15402	0.20907	0.16437	0.09327	0.13046
DCC	0.07381	0.07504	0.13069	0.14250	0.11383	0.18803	0.16262	0.22464	0.17623	0.09921	0.13866
SPM	0.06728	0.07389	0.12678	0.13903	0.10896	0.18433	0.15788	0.21746	0.16914	0.09630	0.13410
NEDI	0.06872	0.07521	0.13116	0.14362	0.11207	0.18867	0.16318	0.22632	0.17614	0.09995	0.13850
ICBI	0.06872	0.07519	0.13104	0.14354	0.11120	0.18860	0.16287	0.22610	0.17347	0.09700	0.13777
SME	0.06820	0.07467	0.12950	0.14171	0.11113	0.18679	0.16153	0.22227	0.17384	0.09823	0.13679
ANR	0.07145	0.07343	0.12578	0.13751	0.11019	0.18304	0.15673	0.21482	0.16838	0.09580	0.13371
RELRL	0.06605	0.07040	0.11617	0.12870	0.10504	0.17372	0.14865	0.19725	0.15678	0.09226	0.12550
Modified RELRL	0.06589	0.06929	0.11335	0.12472	0.10401	0.17166	0.14571	0.19351	0.15462	0.09126	0.12340

Figures 4.1 and 4.2 provide subjective evaluation for both S_0 and $DoLP$ image products. Image 1 is shown for S_0 and Image 2 is used for $DoLP$ comparisons. In both cases, the modified RELRL recovers more edge details and preserves polarization artifacts the best. The RELRL and MKL provide good results as well, however, the RELRL does a better job at recovering HF content compared to MKL.

(a) HR S_0 Image 1

(b) BLI S_0 Image 1

(c) BCI S_0 Image 1

(d) UM S_0 Image 1

(e) MKL S_0 Image 1

(f) DCC S_0 Image 1

(g) SPM S_0 Image 1

(h) NEDI S_0 Image 1

(i) ICBI S_0 Image 1

(j) SME S_0 Image 1

(k) ANR S_0 Image 1

(l) RELR S_0 Image 1

(m) Modified RELR S_0 Image 1

Figure 4.1: Truth scenario, $\times 2$ magnification comparison for S_0

(a) HR *DoLP* Image 2

(b) BLI *DoLP* Image 2

(c) BCI *DoLP* Image 2

(d) UM *DoLP* Image 2

(e) MKL *DoLP* Image 2

(f) DCC *DoLP* Image 2

(g) SPM *DoLP* Image 2

(h) NEDI *DoLP* Image 2

(i) ICBI *DoLP* Image 2

(j) SME *DoLP* Image 2

(k) ANR *DoLP* Image 2

(l) RELRL *DoLP* Image 2

(m) Modified RELRL *DoLP* Image 2

Figure 4.2: Truth scenario, $\times 2$ magnification comparison for *DoLP*

4.3.2 Results for $\times 4$ Magnification

The results in this section are similar to the previous, for the algorithms tested. Note that the modified RELR, DCC, NEDI, and SME did not offer a resolution increase of 4, and therefore are not tested. The same procedure in the previous section is performed here and the results are identical. Tables 4.7 and 4.8 demonstrate SSIM results for the S_0 and $DoLP$ polarimetric image products. The RELR outperforms all other techniques by a slimmer margin for the S_0 image product than the $\times 2$ magnification factor tested in the previous section. The second best performing algorithm is MKL, followed closely by ANR. For the $DoLP$ image product, MKL performed the best, whereas, the RELR performed second best on the testing images.

Table 4.7: SSIM S_0 , Truth Scenario, $\times 4$ Magnification

BLI	0.62043	0.64687	0.77343	0.63211	0.63056	0.65065	0.55195	0.63318	0.48909	0.64122	0.62695
BCI	0.65607	0.68034	0.80038	0.67050	0.66303	0.68341	0.60766	0.69278	0.53117	0.67263	0.66580
UM	0.66425	0.68807	0.80611	0.67924	0.67046	0.69047	0.62018	0.70469	0.54149	0.67910	0.67441
MKL	0.68046	0.70305	0.81734	0.69603	0.68487	0.70514	0.64426	0.72699	0.56081	0.69388	0.69128
SPM	0.65861	0.68386	0.80181	0.67148	0.66547	0.68659	0.60932	0.69528	0.53419	0.67616	0.66828
ICBI	0.60119	0.63308	0.75385	0.60889	0.61098	0.62620	0.53454	0.59160	0.46926	0.61936	0.60489
ANR	0.67511	0.69789	0.80620	0.68614	0.67892	0.70126	0.64011	0.72696	0.55774	0.69040	0.68607
RELR	0.68272	0.70499	0.81670	0.69784	0.68712	0.70488	0.64982	0.72413	0.56826	0.69308	0.69295

Table 4.8: SSIM $DoLP$, Truth Scenario, $\times 4$ Magnification

SR Algorithm	Image 1	Image 2	Image 3	Image 4	Image 5	Image 6	Image 7	Image 8	Image 9	Image 10	Average
BLI	0.53587	0.52865	0.58502	0.47341	0.41184	0.28450	0.34393	0.18030	0.25502	0.57393	0.41725
BCI	0.55238	0.53892	0.60362	0.49232	0.42900	0.29758	0.37268	0.21845	0.27637	0.59143	0.43727
UM	0.55685	0.54194	0.60927	0.49778	0.43399	0.30112	0.38071	0.22929	0.28235	0.59484	0.44282
MKL	0.56638	0.54919	0.62114	0.50983	0.44485	0.31217	0.39758	0.25603	0.29533	0.60618	0.45587
SPM	0.55407	0.54232	0.60083	0.48791	0.43028	0.29955	0.36940	0.21888	0.27526	0.59278	0.43713
ICBI	0.52411	0.52602	0.57965	0.46254	0.40264	0.27785	0.33349	0.16997	0.24374	0.55848	0.40785
ANR	0.55834	0.54818	0.61501	0.49901	0.43841	0.31079	0.39085	0.25732	0.29060	0.59876	0.45073
RELR	0.56333	0.54783	0.61900	0.50807	0.44206	0.30997	0.39472	0.24329	0.29281	0.60298	0.45241

Tables 4.9 and 4.10 indicate that MKL performed the best, followed by the RELR for the S_0 image product. This is different from the $\times 2$ case, as the RELR was the best performer. For the $DoLP$ image product, MKL performs the best and RELR performs second best on the set of testing images.

Table 4.9: PSNR S_0 , Truth Scenario, $\times 4$ Magnification

SR Algorithm	Image 1	Image 2	Image 3	Image 4	Image 5	Image 6	Image 7	Image 8	Image 9	Image 10	Average
BLI	26.38188	25.27959	27.35259	24.19906	25.66192	25.65990	22.44141	24.33756	23.67350	24.54687	24.95343
BCI	27.17559	26.04553	28.31695	25.05273	26.49388	26.63118	23.41191	25.56638	24.33831	25.51523	25.85477
UM	27.33168	26.19543	28.49669	25.21776	26.65893	26.80980	23.60994	25.79544	24.47041	25.69831	26.02844
MKL	27.63899	26.48964	28.86147	25.53266	26.99121	27.17471	23.99779	26.21466	24.72169	26.10726	26.37301
SPM	27.21817	26.05853	28.36407	25.06167	26.51874	26.65711	23.46263	25.61216	24.37011	25.53491	25.88581
ICBI	25.41273	24.49275	26.27372	23.20890	24.64733	24.45153	22.12957	22.79183	23.03961	23.28100	23.97290
ANR	23.11675	20.80339	21.02013	19.40422	22.41025	22.72481	20.25352	24.09183	21.25286	22.21601	21.72938
RELR	27.64225	26.42220	28.85509	25.49529	27.01236	27.16233	24.02527	26.07201	24.71779	26.20426	26.36089

Table 4.10: PSNR $DoLP$, Truth Scenario, $\times 4$ Magnification

SR Algorithm	Image 1	Image 2	Image 3	Image 4	Image 5	Image 6	Image 7	Image 8	Image 9	Image 10	Average
BLI	22.89777	21.95674	16.83487	16.00506	18.56358	13.92024	15.10394	11.86797	14.24246	19.088104	17.04807
BCI	23.08253	22.29112	17.36452	16.59831	18.79623	14.27339	15.53394	12.52019	14.84443	19.629035	17.49337
UM	23.11248	22.37304	17.51104	16.75213	18.83882	14.35245	15.64336	12.69319	14.9905	19.646653	17.59137
MKL	23.20314	22.53251	17.78785	17.02442	18.96428	14.53589	15.84154	13.03074	15.24287	19.832888	17.79961
SPM	23.07008	22.27547	17.33318	16.5669	18.76734	14.25188	15.50097	12.51045	14.81663	19.596059	17.4689
ICBI	22.73038	22.08382	16.99849	16.18952	18.37223	14.01068	15.06398	12.11374	14.24271	18.977605	17.07832
ANR	22.14509	22.52348	17.71848	17.01622	18.6179	14.50084	15.79269	13.03267	15.14773	19.661762	17.61569
RELR	23.14944	22.51565	17.75137	16.97071	18.94546	14.50925	15.78394	12.86751	15.14265	19.916291	17.75523

Finally, Tables 4.11 and 4.12 demonstrate that for the S_0 image product, the RELR and MKL provide the least amount of error per pixel on average, with MKL being the best and RELR being second best. As expected, due to the PSNR performance, MKL provides the least error per pixel and the RELR provides the second least for $DoLP$ imagery.

Table 4.11: RMSE S_0 , Truth Scenario, $\times 4$ Magnification

SR Algorithm	Image 1	Image 2	Image 3	Image 4	Image 5	Image 6	Image 7	Image 8	Image 9	Image 10	Average
BLI	0.04796	0.05445	0.04289	0.06167	0.05211	0.05212	0.07550	0.06069	0.06551	0.05925	0.05721
BCI	0.04377	0.04986	0.03838	0.05589	0.04735	0.04661	0.06752	0.05268	0.06069	0.05300	0.05157
UM	0.04299	0.04900	0.03760	0.05484	0.04646	0.04566	0.06599	0.05131	0.05977	0.05189	0.05055
MKL	0.04150	0.04737	0.03605	0.05289	0.04471	0.04378	0.06311	0.04890	0.05807	0.04950	0.04859
SPM	0.04356	0.04978	0.03818	0.05584	0.04721	0.04647	0.06712	0.05241	0.06046	0.05288	0.05139
ICBI	0.05362	0.05962	0.04856	0.06911	0.05856	0.05990	0.07826	0.07251	0.07047	0.06854	0.06392
ANR	0.06985	0.09117	0.08892	0.10710	0.07577	0.07307	0.09712	0.06243	0.08657	0.07748	0.08295
RELR	0.04148	0.04774	0.03608	0.05312	0.04460	0.04384	0.06291	0.04970	0.05809	0.04895	0.04865

Table 4.12: RMSE $DoLP$, Truth Scenario, $\times 4$ Magnification

SR Algorithm	Image 1	Image 2	Image 3	Image 4	Image 5	Image 6	Image 7	Image 8	Image 9	Image 10	Average
BLI	0.07163	0.07983	0.14396	0.15840	0.11798	0.20137	0.17571	0.25504	0.19403	0.11107	0.15090
BCI	0.07013	0.07681	0.13545	0.14794	0.11487	0.19334	0.16723	0.23659	0.18104	0.10436	0.14278
UM	0.06988	0.07609	0.13318	0.14534	0.11430	0.19159	0.16513	0.23192	0.17802	0.10415	0.14096
MKL	0.06916	0.07471	0.12901	0.14086	0.11266	0.18759	0.16141	0.22308	0.17292	0.10194	0.13733
SPM	0.07023	0.07695	0.13594	0.14848	0.11525	0.19382	0.16786	0.23685	0.18162	0.10476	0.14318
ICBI	0.07303	0.07867	0.14128	0.15507	0.12061	0.19928	0.17652	0.24792	0.19403	0.11249	0.14989
ANR	0.07812	0.07479	0.13004	0.14099	0.11725	0.18835	0.16232	0.22303	0.17483	0.10397	0.13937
RELR	0.06959	0.07485	0.12955	0.14173	0.11291	0.18816	0.16248	0.22731	0.17493	0.10097	0.13825

Figures 4.3 and 4.4 provide subjective evaluation for both S_0 and $DoLP$ image products. Image 1 is shown for S_0 and Image 2 is used for $DoLP$ comparisons. In both cases, the RELR recovers more edge details and preserves polarization artifacts the best, just as in the $\times 2$ case. However, this time ANR provides better edge detail than MKL and offers competitive results with the RELR.

Figure 4.3: Truth scenario, $\times 4$ magnification comparison for S_0

Figure 4.4: Truth scenario, $\times 4$ magnification comparison for *DoLP*

4.4 Derived Microgrid Polarimetric Imagery

4.4.1 Results for $\times 2$ Magnification

Tables 4.13 and 4.14 provides quantitative analysis for the derived microgrid scenario via SSIM of S_0 and *DoLP* polarimetric image products. The modified RELR outperforms all other techniques for both image products. However, it should be noted that the values for all the quantitative

evaluations are lower than its truth scenario counterpart. This is due to the derived microgrid and application of the adaptive demosaicing strategy implemented in [2]. This introduces more aliasing than simply blurring, downsampling, and upsampling HR imagery. The second best algorithm in both cases is the RELR, followed by MKL.

Table 4.13: SSIM S_0 , Derived Microgrid Scenario, $\times 2$ Magnification

SR Algorithm	Image 1	Image 2	Image 3	Image 4	Image 5	Image 6	Image 7	Image 8	Image 9	Image 10	Average
BLI	0.65770	0.67998	0.80093	0.67103	0.66540	0.68696	0.60784	0.69185	0.53431	0.67510	0.66711
BCI	0.68459	0.70533	0.82192	0.70057	0.69072	0.71269	0.64791	0.73084	0.56773	0.69926	0.69616
UM	0.69975	0.71958	0.83336	0.71706	0.70503	0.72700	0.67015	0.75179	0.58704	0.71272	0.71235
MKL	0.70326	0.72261	0.83585	0.72060	0.70802	0.73053	0.67504	0.75578	0.59120	0.71667	0.71596
DCC	0.66476	0.68686	0.80326	0.67901	0.67163	0.69272	0.62437	0.70370	0.54555	0.68079	0.67527
SPM	0.68984	0.71018	0.82613	0.70608	0.69566	0.71767	0.65503	0.73716	0.57418	0.70392	0.70159
NEDI	0.66197	0.68441	0.80344	0.67544	0.66906	0.68991	0.61567	0.69642	0.54016	0.67806	0.67145
ICBI	0.66765	0.69024	0.80832	0.68201	0.67477	0.69554	0.62369	0.70288	0.54710	0.68338	0.67756
SME	0.67629	0.69811	0.81562	0.69099	0.68305	0.70372	0.63603	0.71559	0.55607	0.69167	0.68671
ANR	0.69184	0.71210	0.82651	0.70886	0.69728	0.71954	0.66036	0.74022	0.57850	0.70524	0.70405
RELR	0.73437	0.75230	0.85888	0.75474	0.73790	0.75809	0.72013	0.79569	0.63187	0.74314	0.74871
Modified RELR	0.74221	0.75951	0.86345	0.76216	0.74470	0.76508	0.73111	0.80376	0.64119	0.75078	0.75639

Table 4.14: SSIM $DoLP$, Derived Microgrid Scenario, $\times 2$ Magnification

SR Algorithm	Image 1	Image 2	Image 3	Image 4	Image 5	Image 6	Image 7	Image 8	Image 9	Image 10	Average
BLI	0.51937	0.53308	0.60425	0.48761	0.41934	0.30708	0.36617	0.20452	0.27326	0.57165	0.42863
BCI	0.51486	0.53407	0.61919	0.49925	0.42813	0.32143	0.38405	0.23282	0.28751	0.57382	0.43951
UM	0.50969	0.53281	0.62823	0.50434	0.43117	0.32937	0.39317	0.24949	0.29461	0.57134	0.44442
MKL	0.50897	0.53412	0.62954	0.50903	0.43541	0.33421	0.39667	0.25664	0.29737	0.57693	0.44789
DCC	0.50618	0.52904	0.61037	0.48902	0.42072	0.31467	0.37099	0.22072	0.27717	0.56562	0.43045
SPM	0.51298	0.53353	0.62167	0.50229	0.43079	0.32422	0.38683	0.23845	0.28983	0.57478	0.44154
NEDI	0.51271	0.53115	0.60804	0.48756	0.42074	0.31142	0.36744	0.21166	0.27530	0.56689	0.42929
ICBI	0.50977	0.53077	0.61048	0.48869	0.42120	0.31345	0.36883	0.21594	0.27643	0.56536	0.43009
SME	0.50815	0.53128	0.61434	0.49241	0.42436	0.31731	0.37421	0.22348	0.27999	0.56852	0.43341
ANR	0.50759	0.53123	0.62375	0.50465	0.43201	0.32816	0.38917	0.24662	0.29172	0.57315	0.44281
RELR	0.49626	0.52939	0.63847	0.50987	0.43347	0.33967	0.40677	0.27238	0.30435	0.56190	0.44925
Modified RELR	0.48918	0.52553	0.64230	0.51651	0.43973	0.34498	0.41085	0.27932	0.30590	0.56322	0.45175

Tables 4.15 and 4.16 indicate that the modified RELR is the best, followed closely by RELR and MKL for the S_0 image product. For the $DoLP$ image product, the modified RELR is the best performing algorithm and the RELR performs second best.

Table 4.15: PSNR S_0 , Derived Microgrid Scenario, $\times 2$ Magnification

SR Algorithm	Image 1	Image 2	Image 3	Image 4	Image 5	Image 6	Image 7	Image 8	Image 9	Image 10	Average
BLI	26.95706	25.77801	27.86751	24.73720	26.31326	26.41925	23.16089	25.28281	24.20075	25.28974	25.60065
BCI	27.40776	26.19136	28.31908	25.17876	26.80582	26.97878	23.75139	26.01788	24.60793	25.86153	26.11203
UM	27.64408	26.39688	28.52121	25.39496	27.06835	27.27667	24.07892	26.43545	24.82656	26.17440	26.38175
MKL	27.67533	26.42952	28.52491	25.41599	27.10532	27.31081	24.11826	26.48547	24.86264	26.24211	26.41703
DCC	24.11689	22.20637	22.35036	20.72383	23.34709	23.44180	21.04394	23.90100	22.01306	22.80629	22.59506
SPM	27.49521	26.27746	28.42352	25.26895	26.90178	27.07664	23.85143	26.12596	24.68421	25.95983	26.20650
NEDI	26.87225	25.68822	27.67149	24.60413	26.22003	26.29757	23.21753	25.16563	24.16776	25.18849	25.50931
ICBI	27.05005	25.90642	28.01478	24.84525	26.40427	26.48193	23.38190	25.27758	24.30533	25.34674	25.70142
SME	27.24543	26.10068	28.27208	25.05315	26.62575	26.71501	23.62781	25.53450	24.45712	25.58184	25.92134
ANR	24.88445	22.90971	23.47274	21.58781	24.22358	24.49536	21.80057	24.98712	22.78905	23.818804	23.49692
RELRL	28.19570	26.86735	29.06554	25.92694	27.74041	28.00603	24.86810	27.40055	25.32451	27.04630	27.04414
Modified RELRL	28.24495	26.83982	28.98156	25.89976	27.80862	28.06678	25.03077	27.65927	25.37207	27.17478	27.10784

Table 4.16: PSNR $DoLP$, Derived Microgrid Scenario, $\times 2$ Magnification

SR Algorithm	Image 1	Image 2	Image 3	Image 4	Image 5	Image 6	Image 7	Image 8	Image 9	Image 10	Average
BLI	22.21674	22.56064	17.20733	16.58596	18.56030	14.33569	15.42432	12.23606	14.81677	19.46651	17.34103
BCI	22.07680	22.73898	17.58763	16.99051	18.67272	14.63750	15.69918	12.66860	15.21517	19.79865	17.60857
UM	21.94380	22.79228	17.83183	17.23562	18.71511	14.83170	15.85329	12.94354	15.44224	19.95407	17.75435
MKL	21.94781	22.81505	17.85272	17.25737	18.75717	14.86224	15.88248	12.97975	15.45236	20.01082	17.78178
DCC	21.66889	22.66002	17.46187	16.87646	18.46908	14.55349	15.55712	12.54437	14.99399	19.66275	17.44480
SPM	22.05503	22.75732	17.64767	17.05190	18.69765	14.68737	15.74212	12.73816	15.27378	19.84525	17.64963
NEDI	22.08515	22.61807	17.34678	16.72314	18.54643	14.44739	15.47840	12.39276	14.90235	19.55052	17.40910
ICBI	22.09340	22.65494	17.42287	16.80813	18.56966	14.51387	15.52535	12.48254	14.98040	19.59852	17.46497
SME	22.06652	22.69379	17.51832	16.91393	18.60887	14.60264	15.59599	12.60801	15.09042	19.71370	17.54122
ANR	21.75436	22.76527	17.71772	17.13886	18.63926	14.75684	15.78993	12.82994	15.31969	19.88137	17.65932
RELRL	21.58037	22.71322	18.27894	17.58774	18.71096	15.13291	16.04922	13.38325	15.73150	19.95007	17.91182
Modified RELRL	21.51297	22.64460	18.38691	17.74803	18.76001	15.15016	16.10490	13.39674	15.70999	19.93492	17.93492

Lastly, Tables 4.17 and 4.18 demonstrate that for the S_0 and $DoLP$ image product, the modified RELRL and the RELRL provide the least amount of error per pixel on average, with the modified RELRL being the best and the RELRL being second best.

Table 4.17: RMSE S_0 , Derived Microgrid Scenario, $\times 2$ Magnification

SR Algorithm	Image 1	Image 2	Image 3	Image 4	Image 5	Image 6	Image 7	Image 8	Image 9	Image 10	Average
BLI	0.04489	0.05142	0.04042	0.05796	0.04834	0.04776	0.06950	0.05443	0.06165	0.05439	0.05308
BCI	0.04262	0.04903	0.03837	0.05509	0.04568	0.04478	0.06493	0.05002	0.05883	0.05092	0.05003
UM	0.04148	0.04788	0.03749	0.05373	0.04432	0.04327	0.06253	0.04767	0.05737	0.04912	0.04849
MKL	0.04133	0.04770	0.03748	0.05360	0.04413	0.04310	0.06224	0.04739	0.05713	0.04874	0.04828
DCC	0.06225	0.07757	0.07629	0.09200	0.06802	0.06728	0.08868	0.06382	0.07931	0.07239	0.07476
SPM	0.04219	0.04854	0.03792	0.05452	0.04518	0.04428	0.06418	0.04940	0.05832	0.05035	0.04949
NEDI	0.04533	0.05195	0.04135	0.05886	0.04887	0.04843	0.06904	0.05517	0.06189	0.05503	0.05359
ICBI	0.04441	0.05066	0.03974	0.05725	0.04784	0.04741	0.06775	0.05447	0.06092	0.05403	0.05245
SME	0.04342	0.04954	0.03858	0.05589	0.04664	0.04616	0.06586	0.05288	0.05986	0.05259	0.05114
ANR	0.05699	0.07153	0.06704	0.08329	0.06149	0.05960	0.08128	0.05632	0.07253	0.06443	0.06745
RELRL	0.03892	0.04536	0.03521	0.05054	0.04102	0.03978	0.05709	0.04266	0.05417	0.04443	0.04492
Modified RELRL	0.03870	0.04550	0.03556	0.05070	0.04070	0.03951	0.05604	0.04140	0.05388	0.04378	0.04458

Table 4.18: RMSE $DoLP$, Derived Microgrid Scenario, $\times 2$ Magnification

SR Algorithm	Image 1	Image 2	Image 3	Image 4	Image 5	Image 6	Image 7	Image 8	Image 9	Image 10	Average
BLI	0.07748	0.07447	0.13792	0.14815	0.11803	0.19196	0.16935	0.24445	0.18162	0.10633	0.14498
BCI	0.07873	0.07295	0.13201	0.14141	0.11651	0.18541	0.16407	0.23258	0.17348	0.10235	0.13995
UM	0.07995	0.07251	0.12835	0.13747	0.11594	0.18131	0.16119	0.22533	0.16900	0.10053	0.13716
MKL	0.07991	0.07232	0.12805	0.13713	0.11538	0.18067	0.16065	0.22439	0.16880	0.09988	0.13672
DCC	0.08252	0.07362	0.13394	0.14328	0.11927	0.18721	0.16678	0.23593	0.17795	0.10396	0.14245
SPM	0.07893	0.07280	0.13110	0.14041	0.11618	0.18435	0.16327	0.23072	0.17231	0.10180	0.13919
NEDI	0.07866	0.07398	0.13573	0.14583	0.11822	0.18951	0.16830	0.24008	0.17984	0.10531	0.14354
ICBI	0.07858	0.07366	0.13454	0.14441	0.11790	0.18806	0.16739	0.23761	0.17823	0.10473	0.14251
SME	0.07883	0.07333	0.13307	0.14266	0.11737	0.18615	0.16604	0.23421	0.17599	0.10335	0.14110
ANR	0.08171	0.07273	0.13005	0.13901	0.11696	0.18288	0.16237	0.22830	0.17140	0.10138	0.13868
RELRL	0.08336	0.07317	0.12191	0.13201	0.11600	0.17513	0.15759	0.21421	0.16347	0.10058	0.13374
Modified RELRL	0.08401	0.07375	0.12041	0.12960	0.11535	0.17478	0.15659	0.21388	0.16387	0.10075	0.13330

Figures 4.5 and 4.6 provide subjective evaluation for both S_0 and $DoLP$ image products. Image 1 is shown for S_0 and Image 2 is used for $DoLP$ comparisons. In both cases, the modified RELRL recovers more edge details and preserves polarization artifacts the best, just as in the truth scenario. However, this time there is a greater presence of aliasing, which can be amplified by the super-resolution algorithms. This is particularly present in the $DoLP$ imagery where the car is heavily distorted. While, not doing much to restore HF components, the modified RELRL does the best job at maintaining contrast between the polarized and unpolarized portions of the image, followed closely by the RELRL.

(a) HR S_0 Image 1

(b) BLI S_0 Image 1

(c) BCI S_0 Image 1

(d) UM S_0 Image 1

(e) MKL S_0 Image 1

(f) DCC S_0 Image 1

(g) SPM S_0 Image 1

(h) NEDI S_0 Image 1

(i) ICBI S_0 Image 1

(j) SME S_0 Image 1

(k) ANR S_0 Image 1

(l) RELR S_0 Image 1

(m) Modified RELR S_0 Image 1

Figure 4.5: Derived microgrid scenario, $\times 2$ magnification comparison for S_0

(a) HR *DoLP* Image 2

(b) BLI *DoLP* Image 2

(c) BCI *DoLP* Image 2

(d) UM *DoLP* Image 2

(e) MKL *DoLP* Image 2

(f) DCC *DoLP* Image 2

(g) SPM *DoLP* Image 2

(h) NEDI *DoLP* Image 2

(i) ICBI *DoLP* Image 2

(j) SME *DoLP* Image 2

(k) ANR *DoLP* Image 2

(l) RELR *DoLP* Image 2

(m) Modified RELR *DoLP* Image 2

Figure 4.6: Derived microgrid scenario, $\times 2$ magnification comparison for *DoLP*

4.4.2 Results for $\times 4$ Magnification

The results in this section are similar to the previous, for the algorithms tested. Note that the modified RELR, DCC, NEDI, and SME did not offer a resolution increase of 4, and therefore are not tested. The same procedure in the previous section is performed here and the aliasing problems encountered previously are amplified. Tables 4.19 and 4.20 demonstrate SSIM results for the S_0 and $DoLP$ polarimetric image products. The RELR outperforms all other techniques by a smaller margin for S_0 than the $\times 2$ magnification factor tested in the previous section. The second best algorithm is MKL. For the $DoLP$ case, MKL performs the best and the RELR places second, followed by an UM.

Table 4.19: SSIM S_0 , Derived Microgrid Scenario, $\times 4$ Magnification

SR Algorithm	Image 1	Image 2	Image 3	Image 4	Image 5	Image 6	Image 7	Image 8	Image 9	Image 10	Average
BLI	0.61128	0.63825	0.76572	0.62233	0.62215	0.64133	0.53824	0.61811	0.47959	0.63171	0.61687
BCI	0.64448	0.66931	0.79095	0.65814	0.65259	0.67206	0.58998	0.67474	0.51844	0.66171	0.65324
UM	0.65220	0.67659	0.79646	0.66636	0.65965	0.67878	0.60187	0.68640	0.52797	0.66814	0.66144
MKL	0.66762	0.69088	0.80745	0.68237	0.67359	0.69291	0.62490	0.70854	0.54595	0.68303	0.67772
SPM	0.64683	0.67265	0.79221	0.65878	0.65483	0.67504	0.59135	0.67707	0.52113	0.66505	0.65549
ICBI	0.59612	0.62747	0.74929	0.60305	0.60631	0.62110	0.52467	0.58346	0.46371	0.61418	0.59894
ANR	0.66278	0.68607	0.79651	0.67286	0.66788	0.68910	0.62105	0.70837	0.54282	0.67973	0.67272
RELR	0.67169	0.69445	0.80853	0.68583	0.67746	0.69423	0.63269	0.70995	0.55429	0.68436	0.68135

Table 4.20: SSIM $DoLP$, Derived Microgrid Scenario, $\times 4$ Magnification

SR Algorithm	Image 1	Image 2	Image 3	Image 4	Image 5	Image 6	Image 7	Image 8	Image 9	Image 10	Average
BLI	0.51324	0.52426	0.58240	0.46619	0.40025	0.28393	0.33197	0.16302	0.24737	0.55511	0.40677
BCI	0.50986	0.52675	0.59933	0.47747	0.40785	0.29710	0.35066	0.19002	0.26171	0.55742	0.41781
UM	0.50666	0.52605	0.60397	0.47994	0.40929	0.30036	0.35479	0.19739	0.26477	0.55537	0.41986
MKL	0.49964	0.52473	0.61232	0.48475	0.41210	0.30879	0.36271	0.21361	0.27064	0.55465	0.42439
SPM	0.51148	0.52975	0.59643	0.47313	0.40939	0.29904	0.34727	0.18995	0.26030	0.55810	0.41748
ICBI	0.48940	0.51574	0.58205	0.45137	0.38785	0.27792	0.31348	0.15502	0.23389	0.52812	0.39349
ANR	0.49260	0.52204	0.60688	0.47352	0.40741	0.30776	0.35592	0.21397	0.26531	0.54604	0.41915
RELR	0.49853	0.52541	0.60950	0.48520	0.41059	0.30635	0.35971	0.20524	0.26842	0.55534	0.42243

Tables 4.21 and 4.22 indicate that the RELR performed the best, followed by MKL for the S_0 image product. For the $DoLP$ image product, MKL performs the best and the RELR performs second best.

Table 4.21: PSNR S_0 , Derived Microgrid Scenario, $\times 4$ Magnification

SR Algorithm	Image 1	Image 2	Image 3	Image 4	Image 5	Image 6	Image 7	Image 8	Image 9	Image 10	Average
BLI	26.17863	25.07796	27.09895	23.98287	25.44722	25.41120	22.20571	24.04336	23.50964	24.28705	24.72426
BCI	26.93022	25.80127	28.00040	24.78862	26.23467	26.31923	23.08926	25.19574	24.13736	25.20448	25.57013
UM	27.08388	25.94903	28.17682	24.95038	26.39712	26.49360	23.28064	25.42351	24.26612	25.38673	25.74078
MKL	27.37792	26.23464	28.52304	25.25080	26.72034	26.83219	23.65000	25.83056	24.50438	25.77186	26.06957
SPM	26.96907	25.81073	28.04002	24.79250	26.25598	26.34001	23.13821	25.23816	24.16535	25.21863	25.59687
ICBI	25.36964	24.42625	26.19585	23.14789	24.59153	24.37900	21.97726	22.75607	22.98284	23.21866	23.90450
ANR	23.02627	20.73646	20.96433	19.33578	22.31514	22.59741	20.11032	23.85046	21.15503	22.07311	21.61643
RELR	27.48384	26.27750	28.65375	25.32321	26.85314	26.94925	23.78973	25.88484	24.57398	26.01658	26.18058

Table 4.22: PSNR $DoLP$, Derived Microgrid Scenario, $\times 4$ Magnification

SR Algorithm	Image 1	Image 2	Image 3	Image 4	Image 5	Image 6	Image 7	Image 8	Image 9	Image 10	Average
BLI	22.24656	22.18489	16.70353	16.00987	18.32426	13.92401	14.98987	11.65058	14.20209	18.86057	16.90962
BCI	22.08872	22.53066	17.19009	16.58539	18.44333	14.31363	15.35851	12.20453	14.77900	19.35520	17.28491
UM	22.00815	22.58083	17.32481	16.72928	18.45729	14.40983	15.44320	12.35403	14.91229	19.39688	17.36166
MKL	21.85925	22.63944	17.56835	16.97090	18.49439	14.59095	15.58269	12.62998	15.11769	19.53262	17.49863
SPM	22.11472	22.51538	17.15636	16.55259	18.41883	14.29357	15.32341	12.19211	14.74861	19.32152	17.26371
ICBI	21.96157	22.29133	16.91525	16.22398	18.13035	14.06315	14.95374	11.90331	14.23832	18.74118	16.94222
ANR	21.19539	22.57682	17.54101	16.97903	18.22801	14.58686	15.54147	12.66401	15.05397	19.40748	17.37740
RELR	21.88025	22.64224	17.51881	16.91910	18.49216	14.52812	15.53068	12.49833	15.01727	19.60130	17.46283

Finally, Tables 4.23 and 4.24 demonstrate that for the S_0 image product, the RELR and MKL provide the least amount of error per pixel on average, with RELR being the best and MKL being second best. As expected, due to the PSNR performance, MKL provides the least error per pixel and the RELR provides the second least for $DoLP$ imagery.

Table 4.23: RMSE S_0 , Derived Microgrid Scenario, $\times 4$ Magnification

SR Algorithm	Image 1	Image 2	Image 3	Image 4	Image 5	Image 6	Image 7	Image 8	Image 9	Image 10	Average
BLI	0.04910	0.05573	0.04416	0.06322	0.05341	0.05363	0.07757	0.06278	0.06676	0.06104	0.05874
BCI	0.04503	0.05128	0.03981	0.05762	0.04878	0.04831	0.07007	0.05498	0.06211	0.05493	0.05329
UM	0.04424	0.05041	0.03901	0.05656	0.04788	0.04735	0.06854	0.05356	0.06119	0.05379	0.05225
MKL	0.04277	0.04878	0.03748	0.05463	0.04613	0.04554	0.06569	0.05111	0.05954	0.05145	0.05031
SPM	0.04483	0.05122	0.03963	0.05759	0.04866	0.04819	0.06968	0.05471	0.06191	0.05484	0.05313
ICBI	0.05389	0.06007	0.04900	0.06960	0.05894	0.06040	0.07964	0.07281	0.07093	0.06903	0.06443
ANR	0.07058	0.09187	0.08949	0.10795	0.07660	0.07415	0.09874	0.06419	0.08755	0.07877	0.08399
RELR	0.04225	0.04854	0.03692	0.05418	0.04543	0.04493	0.06464	0.05079	0.05906	0.05002	0.04968

Table 4.24: RMSE $DoLP$, Derived Microgrid Scenario, $\times 4$ Magnification

SR Algorithm	Image 1	Image 2	Image 3	Image 4	Image 5	Image 6	Image 7	Image 8	Image 9	Image 10	Average
BLI	0.07721	0.07776	0.14616	0.15831	0.12128	0.20128	0.17804	0.26150	0.19494	0.11402	0.15305
BCI	0.07863	0.07473	0.13820	0.14816	0.11963	0.19245	0.17064	0.24534	0.18241	0.10771	0.14579
UM	0.07936	0.07429	0.13607	0.14573	0.11944	0.19033	0.16898	0.24116	0.17963	0.10719	0.14422
MKL	0.08073	0.07380	0.13231	0.14173	0.11893	0.18640	0.16629	0.23362	0.17543	0.10553	0.14148
SPM	0.07839	0.07486	0.13873	0.14872	0.11997	0.19290	0.17133	0.24569	0.18305	0.10812	0.14618
ICBI	0.07979	0.07681	0.14264	0.15445	0.12402	0.19808	0.17878	0.25400	0.19413	0.11560	0.15183
ANR	0.08714	0.07433	0.13272	0.14160	0.12263	0.18649	0.16708	0.23270	0.17673	0.10706	0.14285
RELR	0.08054	0.07377	0.13306	0.14258	0.11896	0.18776	0.16729	0.23718	0.17747	0.10470	0.14233

Figures 4.7 and 4.8 provide subjective evaluation for both S_0 and $DoLP$ image products. Image 1 is shown for S_0 and Image 2 is used for $DoLP$ comparisons. For the S_0 case, the RELR recovers more edge details. As for the $DoLP$ case, none of the super-resolution algorithms tested could overcome the aliasing in order to provide good subjective results. However, the RELR preserves polarization contrast the best, just as in the truth scenario.

Figure 4.7: Derived microgrid scenario, $\times 4$ magnification comparison for S_0

Figure 4.8: Derived microgrid scenario, $\times 4$ magnification comparison for *DoLP*

4.5 Runtime Comparison

For the purposes of real-time applications, an appropriate tradeoff between performance and speed is an important decision. Simple interpolation algorithms, such as BLI and BCI are fast but provide blurry images with no attempt at restoring HF components. Super-resolution algorithms, such as those tested in this thesis, very greatly on performance and speed. Figures 4.25 and 4.26

demonstrated that when it comes to subjective evaluation and speed, the modified RELR wins in the subjective evaluation but loses to the RELR in speed. While MKL gives the third best results on average, it is much slower than ANR and an unsharp mask that offer competitive results. ICBI, while the fastest of the super-resolution algorithms did not provide results that were better than BCI or BLI. SPM, SME, and NEDI all provide stable results but take far too long to complete to be useful for real-time applications. DCC was the fourth fastest super-resolution algorithm but did not perform as well as ANR subjectively or objectively. For the $\times 2$ magnification, the RELR outperformed its counterparts by as much as 440 times to as little as 1.6 times. This is with exception to ICBI as mentioned earlier. For the $\times 4$ case, the RELR drops in speed due to the increase in number of hidden nodes, but is still 2 times to 31 times as fast as its counterparts. This is again with exception to ICBI.

Table 4.25: Run time in seconds with modified RELR as reference , $\times 2$ magnification

Run Time	Average	Performance Difference
BLI	0.010	0.002
BCI	0.010	0.002
UM	0.014	0.003
MKL	35.219	7.525
DCC	8.931	1.908
SPM	97.582	20.851
NEDI	73.857	15.781
ICBI	1.540	0.329
SME	1240.000	264.957
ANR	10.077	2.153
RELR	2.814	0.601
Modified RELR	4.680	1

Table 4.26: Run time in seconds with RELR as reference, $\times 4$ magnification

Run Time	Average	Performance Difference
BLI	0.010	0.003
BCI	0.010	0.003
UM	0.019	0.006
MKL	33.914	10.976
SPM	98.750	31.961
ICBI	2.157	0.698
ANR	5.964	1.930
RELR	3.090	1.000

4.6 Discussion

Several single-image super-resolution techniques were compared quantitatively and subjectively at different magnification factors for microgrid polarimetric data. In terms of speed, simple interpolation strategies were the fastest but the least effective in recovering HF components. Surprisingly, application of an unsharp mask proved to be more effective than most of the super-resolution techniques applied throughout the paper. The apparent under performance of DCC, NEDI, ICBI, and SME, when compared to BCI and BLI, can be justified after further observing how these algorithms performed in other papers. For example, in [4] these same algorithms also did not perform as well as the simple interpolation algorithms. This could be due to the algorithms being optimized for a particular data set and not having optimum parameters.

Starting with S_0 , the modified RELR gave the best SSIM, PSNR, and RMSE scores across the board for $2\times$ magnification. Subjectively, the modified RELR also did a better job at recovering HF components on edges and making the images look sharper. Looking at the *DoLP*, the modified RELR algorithm gave the best performance for SSIM, PSNR, and RMSE. When subjectively comparing the *DoLP* in the Derived Microgrid Scenarios, it became apparent that aliasing greatly distorted the images for magnification of both $\times 2$ and $\times 4$, which hindered the performance of all algorithms. None of the super-resolution methods could do much to correct for the aliasing and thus

added HF components to aliased parts of the image, further distorting it. This indicates that work needs to be done in order to combat aliasing and will be the focus of future work.

In general, the modified RELR performed better than all other algorithms. Subjectively when comparing the reconstructed image to the HR image, many polarization artifacts are blurred in the other algorithms but still present in the reconstructed image produced by the modified RELR. The higher SSIM indicates this too, as the modified RELR performs better than all other algorithms in every testing scenario except for the $\times 4$ magnification, where it did not offer a solution. Figures 4.9 and 4.10 demonstrate how the modified RELR introduces less noise and recovers more details than its RELR counterpart.

Figure 4.9: Truth scenario, $\times 2$ magnification comparison for S_0

Figure 4.10: Derived microgrid scenario, $\times 2$ magnification comparison for S_0

When it comes to speed, the RELR when compared to the other single-image super-resolution algorithms, was by far the fastest (ranging from 1.6 to 440 times better performance) with the exception of ICBI. Although ICBI was faster than the RELR, it was also one of the poorest performing techniques for this particular type of data. The modified RELR, although slower than the RELR and ICBI, was still faster than all other single-image super-resolution algorithms (ranging from 2 to 265 times better performance). It should also be noted the simplicity of implementation of the RELR and the modified RELR allows for optimization through parallel programming. This makes greater performance with a reduction in speed possible since the end test image is produced by propagating features of the image through a series of multiplication and additions that can be performed in parallel on a GPU. This means a greater number of hidden nodes can be used without a reduction in speed, thus resulting in better prediction and recovery of HF and LF components. Considering all of this, although the modified RELR is slower than the RELR, the difference will be minimal when it is GPU optimized. Overall, the modified RELR outperformed the other super-resolution algorithms

and is exceptionally simple to implement and scale. This provides great opportunity for real-time applications where speed is a priority.

CHAPTER V

CONCLUSION

Demosaicing strategies for microgrid polarimeters suffer from undersampling. This results in aliasing and a loss in HF content. The modified RELR proposed in this thesis minimizes these effects compared to other state-of-the-art methods by preserving LF content before attempting to recover HF details. The modified RELR outperformed its single-image super-resolution counterparts in terms of SSIM, PSNR, RMSE, and subjective evaluation for the S_0 and $DoLP$ image products.

The modified RELR incorporates a LF correction step before attempting to recover HF components. This step was crucial to removing noise and recovering finer details and structures from within LR images. While slower than the RELR, due to its simplicity and ability to be optimized through parallel programming, the actual cost in speed is negligible. This provides a foundation for microgrid polarizers where super-resolution in real-time applications is desired.

The simulation results for $\times 2$, and especially $\times 4$ magnification factors, demonstrate the need for combating aliasing. Aliasing artifacts were less prominent in the modified RELR but still proved to be a problem for the $DoLP$ image product. Future work is dedicated to addressing aliasing in terms of a better super-resolution algorithm or demosaicing process through machine learning processes.

BIBLIOGRAPHY

- [1] J. S. Tyo, D. L. Goldstein, D. B. Chenault, and J. A. Shaw, "Review of passive imaging polarimetry for remote sensing applications," *Applied optics*, vol. 45, no. 22, pp. 5453–5469, 2006.
- [2] B. M. Ratliff, C. F. LaCasse, and J. S. Tyo, "Adaptive strategy for demosaicing microgrid polarimeter imagery," in *Aerospace Conference, 2011 IEEE*. IEEE, 2011, pp. 1–9.
- [3] R. C. Hardie, D. A. LeMaster, and B. M. Ratliff, "Super-resolution for imagery from integrated microgrid polarimeters," *Optics express*, vol. 19, no. 14, pp. 12 937–12 960, 2011.
- [4] P. Sidike, E. Krieger, M. Z. Alom, V. K. Asari, and T. Taha, "A fast single-image super-resolution via directional edge-guided regularized extreme learning regression," *Signal, Image and Video Processing*, pp. 1–8, 2017.
- [5] D. Zhou, X. Shen, and W. Dong, "Image zooming using directional cubic convolution interpolation," *IET image processing*, vol. 6, no. 6, pp. 627–634, 2012.
- [6] X. Hu, S. Peng, and W.-L. Hwang, "Learning adaptive interpolation kernels for fast single-image super resolution," *Signal, Image and Video Processing*, vol. 8, no. 6, pp. 1077–1086, 2014.
- [7] X. Li and M. T. Orchard, "New edge-directed interpolation," *IEEE transactions on image processing*, vol. 10, no. 10, pp. 1521–1527, 2001.
- [8] A. Giachetti and N. Asuni, "Real-time artifact-free image upscaling," *IEEE Transactions on Image Processing*, vol. 20, no. 10, pp. 2760–2768, 2011.
- [9] S. Mallat and G. Yu, "Super-resolution with sparse mixing estimators," *IEEE Transactions on Image Processing*, vol. 19, no. 11, pp. 2889–2900, 2010.
- [10] T. Peleg and M. Elad, "A statistical prediction model based on sparse representations for single image super-resolution," *IEEE transactions on image processing*, vol. 23, no. 6, pp. 2569–2582, 2014.

- [11] R. Timofte, V. De Smet, and L. Van Gool, “Anchored neighborhood regression for fast example-based super-resolution,” in *Proceedings of the IEEE International Conference on Computer Vision*, 2013, pp. 1920–1927.
- [12] K. S. Ni, S. Kumar, N. Vasconcelos, and T. Q. Nguyen, “Single image superresolution based on support vector regression,” in *Acoustics, Speech and Signal Processing, 2006. ICASSP 2006 Proceedings. 2006 IEEE International Conference on*, vol. 2. IEEE, 2006, pp. II–II.
- [13] K. I. Kim and Y. Kwon, “Example-based learning for single-image super-resolution,” in *Joint Pattern Recognition Symposium*. Springer, 2008, pp. 456–465.
- [14] M.-C. Yang and Y.-C. F. Wang, “A self-learning approach to single image super-resolution,” *IEEE Transactions on Multimedia*, vol. 15, no. 3, pp. 498–508, 2013.
- [15] L. An and B. Bhanu, “Image super-resolution by extreme learning machine,” in *Image processing (ICIP), 2012 19th IEEE International Conference on*. IEEE, 2012, pp. 2209–2212.
- [16] G.-B. Huang, H. Zhou, X. Ding, and R. Zhang, “Extreme learning machine for regression and multiclass classification,” *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, vol. 42, no. 2, pp. 513–529, 2012.
- [17] G.-B. Huang, L. Chen, C. K. Siew *et al.*, “Universal approximation using incremental constructive feedforward networks with random hidden nodes,” *IEEE Trans. Neural Networks*, vol. 17, no. 4, pp. 879–892, 2006.
- [18] G.-B. Huang and L. Chen, “Convex incremental extreme learning machine,” *Neurocomputing*, vol. 70, no. 16, pp. 3056–3062, 2007.
- [19] —, “Enhanced random search based incremental extreme learning machine,” *Neurocomputing*, vol. 71, no. 16, pp. 3460–3468, 2008.
- [20] A. E. Hoerl and R. W. Kennard, “Ridge regression: Biased estimation for nonorthogonal problems,” *Technometrics*, vol. 12, no. 1, pp. 55–67, 1970.
- [21] Z. Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli, “Image quality assessment: from error visibility to structural similarity,” *IEEE transactions on image processing*, vol. 13, no. 4, pp. 600–612, 2004.