

# Ensemble CNNs in Transform Domains for Small Data Image Super-resolution

A Dissertation

Submitted to the Faculty

of the

WORCESTER POLYTECHNIC INSTITUTE

In partial fulfillment of the requirements for the

Degree of Doctor of Philosophy

in

Data Science

Yingnan Liu

Thursday 11<sup>th</sup> August, 2022

**Committee Members:**

Prof. Randy C. Paffenroth, Worcester Polytechnic Institute. Advisor.

Prof. Mohamed Y. Eltabakh, Worcester Polytechnic Institute

Prof. Yanhua Li, Worcester Polytechnic Institute

Dr. Rachel Stephenson, Nanocomp Technologies, Inc.

# Contents

|          |   |           |
|----------|---|-----------|
| <b>1</b> | <b>Introduction</b>   | <b>12</b> |
| <b>2</b> | <b>Related Works</b>  | <b>17</b> |
| 2.1      | Compressive Sensing . . . . .   | 17        |
| 2.2      | Transform Domains . . . . .   | 18        |
| 2.3      | Super-resolution CNNs . . . . .   | 23        |
| <b>3</b> | <b>Methodology for Data Science</b>   | <b>26</b> |
| 3.1      | The comparison between architectures of different algorithms . . . . .  | 28        |
| 3.2      | SparseFnet: Shallow Super-resolution CNN in the Frequency domain . . . . .  | 32        |
| 3.3      | ESnet: Ensemble Super-resolution for CNNs in Transform Domains . . . . .  | 39        |
| 3.4      | EWnet: Ensemble Super-resolution CNN in Wavelet domains . . . . .   | 51        |
| 3.5      | EnsemNet: Ensemble Super-resolution CNN in Wavelet domains for Small Data and Diverse Compressive Models . . . . .  | 59        |
| 3.6      | Voxel-based: Multi-layer Wavelet Transformations for Image Super-resolution: Applications to Voxel-based Deep Learning and Areal Density Maps of Carbon-nanotube Sheets . . . . . | 66        |
| <b>4</b> | <b>Application of MIRALON Areal Density Maps</b>  | <b>67</b> |
| 4.1      | Super-resolution Neural Network with shim-stock ground truth . . . . .  | 69        |
| 4.1.1    | Reconstruction of MIRALON Areal Density Maps with SparseFnet . . . . .  | 69        |
| 4.1.2    | Reconstruction of MIRALON Areal Density Maps with EWnet . . . . .   | 73        |
| 4.1.3    | Reconstruction of MIRALON Areal Density Maps with EnsemNet . . . . .  | 76        |
| 4.2      | Semi-3D Super-resolution with MIRALON Sheet ground truth . . . . .  | 78        |
| 4.2.1    | 2D super resolution methods . . . . .   | 81        |
| 4.2.2    | Voxel construction . . . . .  | 81        |
| 4.2.3    | The rotation of voxel patches . . . . .   | 83        |
| 4.2.4    | Semi-3D Super-resolution result . . . . .   | 84        |
| <b>5</b> | <b>Future Work</b>  | <b>91</b> |
| <b>6</b> | <b>Conclusion</b>   | <b>95</b> |
|          | <b>Appendices</b>   | <b>96</b> |

## Abstract

In this research, we develop methods for single image super-resolution by combining ideas inspired by compressive sensing with super-resolution neural networks and ensemble learning. We are interested in problems where large data sets are not available such as healthcare and aerospace applications. The essence of our work is to use ideas similar to those leveraged by compressive sensing, namely sparse representations, for a robust model accommodating small training data. We develop and demonstrate techniques for combining classic sparse representations with modern ideas in deep neural networks, such as neural network ensembles, to improve the performance of image super-resolution task. Particularly, we report here a successful application of our model to improve the resolution of areal density maps of carbon nanotube sheets generated by a beta particle transmission system. We show that applying our models can reveal finer details in the material texture, helping to improve the detection of manufacturing defects and improving the quality control capabilities for carbon nanotube sheet production.

## Acknowledgments

I thank my advisor, Prof. Randy Paffenroth, for supporting me during these past five years of the academic journey pursuing the Ph.D. degree as well as the previous two years of guiding me into the data science research field. Prof. Paffenroth advised me to step into the doctoral studies. I would never make the decision to become a full time Ph.D. student in WPI without his inspiration and encouragement. Getting along with him in nearly a decade, Prof. Paffenroth keeps giving me recognition. I gained confidence in not only academic studies but also my future careers. He has been a mentor to me since the first day that I attended his class eight years ago.

This research is sponsored by Nanocomp Technologies Inc. I thank the team at Nanocomp for their assistance and data generation to support this work. I would like to give special thanks to quality assurance manager Bob Casoni for the efficient communication between manufacturing questions and machine learning tasks, flexible time schedule for industrial deliverable and my research requirements and supports on accommodating software and platforms. I would also like to express my appreciation to research scientist Dr. Rachel Stephenson who inspired me with valuable ideas, making the project possible on bridging the carbon nano-tube manufacturing task with image super-resolution research. Last but not the least, I would like to give thanks to quality engineer Nick Dixon who spent a lot time on generating experimental data and providing the fundamental data support.

I would like to thank my committee advisors, WPI Profs. Mohamed Eltabakh and Yanhua Li, for the time they have invested in reading my manuscripts and providing valuable feedback.

## Paper Contribution

**Liu, Y.**, Paffenroth, R. C. CNNs in the frequency domain for image super-resolution. (2019). International Conference on Image and Video Processing, and Artificial Intelligence, 11321, 113210D.

**Liu, Y.**, Paffenroth, R. C. Ensemble CNN in Transform Domains for Image Super-resolution from Small Data Sets. (2020)). International Conference on Machine Learning and Applications, 384-391.

**Liu, Y.**, Paffenroth, R. C. Voxel based Deep Learning for Image Super-resolution of Areal Density Maps of Carbon nanotube Sheets. (2021). International Conference on Machine Learning and Applications.

**Liu, Y.**, Paffenroth, R. C. Multi-layer Wavelet Transformations for Image Super-resolution: Applications to Voxel-based Deep Learning and Areal Density Maps of Carbon-nanotube Sheets (2022). Deep Learning Applications, Volume 4.

Submitted: **Liu, Y.**, Paffenroth, R. C. Ensemble Image Super-resolution CNNs for Small Data and Diverse Compressive Models. (2022). International Conference Advanced Data Mining and Applications.

# Executive Summary

Our work has been inspired by a real-world material science problem, and has been funded through the generosity of Nanocomp Technologies, Inc. The application goal of this research is to deliver high-resolution areal density maps of Nanocomp’s MIRALON carbon nanotube sheet material with minimized time required to scan the articles to aid in quality control for the manufacturing process. From the perspective of data science, this is equivalent to a single image super-resolution problem. However, there are three major challenges that arise from the specific application that make this research stand apart from other algorithms. First, there were no high-resolution ground truth scans available for supervised training based algorithms. Our collaborators from Nanocomp were able to provide **a small collection of training data** for the neural work. However, the data is far from the required amount that can be sufficient for the deep image super-resolution neural works, which is not uncommon for machine learning problems in the physical sciences. **Accordingly, one of the key contributions of this thesis is the development of image super-resolution algorithms that are effective in the low-data limit.** Second, unlike the camera measurements for standard visual imagery, we not only need to determine the image compressive measurement from the mathematical functions in the camera, but also the emitter trajectory and beta particles’ behavior from the beta transmission equipment. **The compressive measurement is very complex** due to the settings on the equipment and, as far as we are aware, **image super-resolution techniques have not been previously applied in such complex measurement environments.** Third, the MIRALON sheet areal density map is a 2-dimensional projection of the density distribution. However, the compressive measurement has **depth dependency in 3-dimensional space** due to the interactions of the beta particles through the thickness of the sheet. **As far as we are aware, this thesis presents the first application of deep learning to volumetric super-resolution problems, especially as applied to real-world materials.**

To address the above challenges with deep super-resolution neural networks, we develop methods by combining ideas inspired by compressive sensing and ensemble learning. The fundamental of our work is to use the idea of classic sparse representations that are leveraged by compressive sensing for a robust model accommodating small training data. We developed five models which are reflected in five papers from this research. The first model is SparseFnet, in which the neural network can only access the information on the Fourier domain. We demonstrate that with the help of sparse representations, the neural network could be sufficiently trained even when the size of the training data is very small. In addition, we indicate that instead of numerous deep layers, a shallower architecture in the frequency

domain is more efficient and can guarantee a more stable performance. The development of this model resulted in *CNNs in the frequency domain for image super-resolution*, published in 2019 International Conference on Image and Video Processing, and Artificial Intelligence. Nevertheless, there are many sparsifying transformations that are competitive choices for image processing problems [1]. As the optimal choice depends on the specific image content, the best-suited transformation may differ for different application fields, different compressive measurements and even different sub-areas of a single image [2]. As shown in Figure 1, the frequency and pixel correlations differ from sub-areas in the same image. Accordingly, the optimal choice of transform domains for sub-patches are different. Therefore, we introduce multiple wavelet transformations into SparseFnet to develop an ensemble model which is called ESnet. ESnet makes multiple high-resolution reconstructions with the basic SparseFnet on different transform domains and then combines the results, delivering an optimal combination of those reconstructions. Our ensemble method has a robust performance on limited available data. The ensemble decides the best choice of transform domains for each sub-image patch and combines the patches to provide a general optimal solution for the entire image. This work is in proceeding in the conference of ICMLA 2020: International Conference on Machine Learning and Applications, as *Ensemble CNN in Transform Domains for Image Super-resolution from Small Data Sets*. Although ESnet significantly improves the performance compared with SparseFnet, it is a 'reconstruct-then-ensemble' 2-stage model. We believe that by simplifying the architecture in a more uniform fashion, the result would be more accurate. There are two models, EWnet and EnsemNet, proposed as improved versions of ESnet from two different aspects. EWnet increases the efficiency by extracting sparse features across different transform domains. As developed here, the model performs across multiple wavelet domains and outputs a high-resolution reconstruction in the space domain. EWnet not only simplifies the training procedure but also further improves the reconstruction quality. Instead of focusing on feature extraction, EnsemNet ensembles reconstructions from benchmark algorithms which are performed on different domains to further improve the robust performance. This work is reflected in the conference paper *Ensemble Image Super-resolution CNNs for Small Data and Diverse Compressive Models* which is submitted to ADMA 2022: International Conference Advanced Data Mining and Applications. The advantages of our innovative models compared with other recent works in the image super-resolution field are demonstrated on standard visual imagery in the 2-dimensional space. After validation, we apply the algorithms to the initial challenge of reconstructing the areal density maps of MIRALON sheets. EnsemNet is the latest version of the algorithms that leverages all the advantage simultaneously, including sparse representations, shallow architectures, high-performance benchmark networks and ensembles. We adapt it into a semi-3D

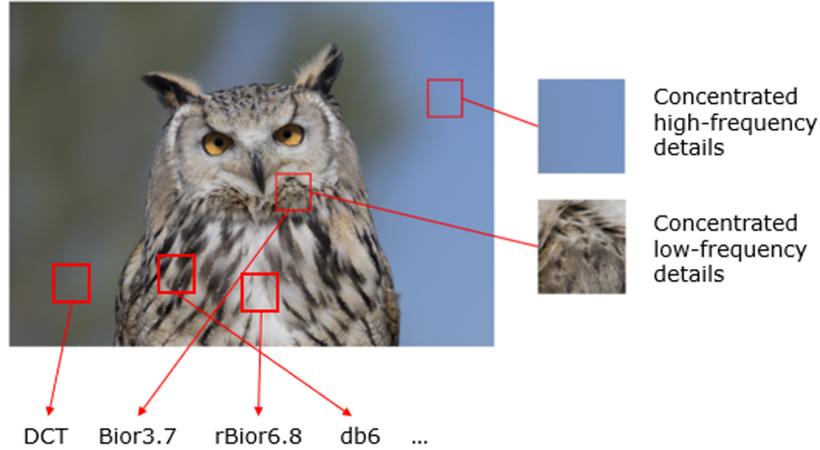


Figure 1: The frequency and pixel correlations differ from sub-areas in the same image. The optimal choice of transform domains for the sub-areas are different.

voxel-based structure to accommodate the depth dependency in beta transmission images.

To separate the two-part challenge of understanding how the instrument processes the collected signal from determining the real variation in areal density of a MIRALON sheet, the initial experiments are conducted on training data collected from a standard shim-stock material. Shim-stock is a readily available plastic material of uniform known thickness that we assume will have a uniform areal density due to how it is made. We can then generate a reasonable set of training data, including scans where known figures and patterns are cut in to it to assess the response of the detector. While still small, this set of training data is larger than would have been available if only the small pieces of uniform MIRALON sheet were used instead of shim-stock. One thing unknown at the start was if the compressive measurement from the instrument would be the same for shim-stock and the MIRALON sheets. Additionally, the beta particles could not be assumed to have a uniform interaction with the sheet (MIRALON or shim-stock) at any height between the source and the detector. To solve these two problems, we extend our work into a semi-3D voxel-based design and involve further data augmentations to enrich the data set. Taking advantage of different transform domains, the ensemble method and additional engineering measurements in semi-3D space, our research reveals finer details in the material texture. Figure 2 shows a preview of the reconstruction result. The defective spots on a MIRALON sheet are easier to identify after super-resolution of 4 times in both the x and y dimension. The semi-3D method is reflected in two papers: (1) *Voxel based Deep Learning for Image Super-resolution of Areal Density Maps of Carbon nanotube Sheets*. in ICMLA 2021, and (2) the book chapter *Multi-layer Wavelet Transformations for Image Super-resolution: Applications to Voxel-based Deep Learning and Areal Density Maps of Carbon-nanotube Sheets*. in *Deep Learning Applications*,

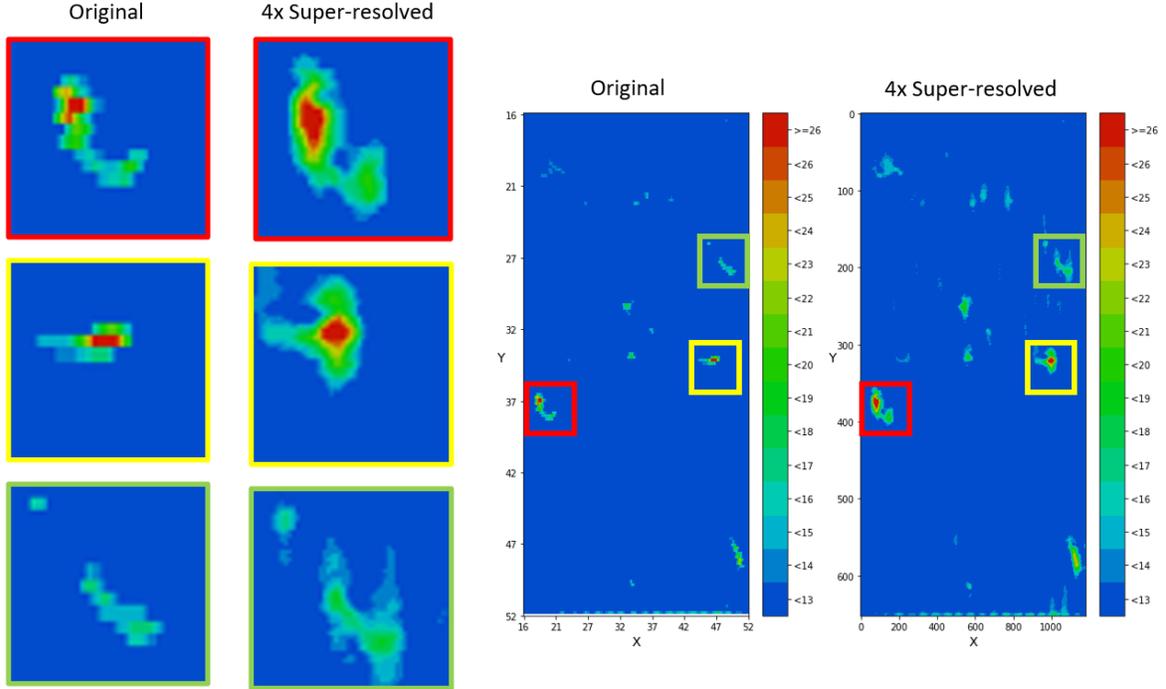


Figure 2: Defective spots on a MIRALON<sup>®</sup> sheet. The image from the original beta transmission areal density maps is heavily down-sampled. The super-resolution results from the proposed experiment recover fine texture details.

Volume 4.

In this work, we have developed image super-resolution neural networks with sparse representatives, the ensemble method and advanced engineering measurements in semi-3D space. Our study stands apart from previous algorithms by adapting different image content, complex compressive measurements and the restriction of small training data. Our work is novel in four ways.

1. First, our method stands on the assumption that high-resolution images can be reconstructed more effectively when the image super-resolution neural network is trained in the transform domain. Even with a shallower architecture, sparse representatives help to capture the features of an image more efficiently than the original space domain. Therefore, we can overcome the challenge of a small amount of training data.
2. Second, the optimal choice of transform domains is not identical for not only different types of images but also different sub-areas in a single image. A universal best choice of transform domain does not exist that can be most effectively processed by CNNs for all types of images and all types of compressive measurements. Our ensemble model learns properties of both the images and transform domains to make the best choice.

3. Third, the best-suited sparsifying transformation may vary from different sub-areas in the entire image and different compressive measurements. Based upon advantageous properties of different transform domains, our ensemble models combine the results from different domains for sub-image patches with different characteristics. A robust solution to image super-resolution is therefore provided.
4. Fourth, the beta transmission areal density map is a 2D projection of the MIRALON sheet where the reported areal density is dependent on where the sheet is placed in the space between the source and the detector. Multiple transform domains, the ensemble method, the advanced engineering measurement and data augmentation allow us to extend the work into 3D space, providing an accurate reconstruction for MIRALON sheet areal density map applications.

# Research Road-map



Figure 3: Summary of our research progress. The orange box states the application problem, goal and the research field of our studies. The progress of theory development between application challenges and the corresponding machine learning solutions are listed. Dark blue boxes indicate the essential challenges that make our study different than ordinary image super-resolution problems. Light blue boxes are intermediate problems which need to be addressed. The novel contributions are highlighted with dark green boxes as key features, while other improvements are shown in light green boxes. The associated publications can be found in the yellow boxes.

# 1 Introduction

Compressive sensing and convolutional neural networks (CNN) are both popular techniques for high-resolution image reconstruction. Compressive sensing is a classical approach which solves the problem by using bases in which signals can be sparsely represented, and such methods allow one to reconstruct the high-resolution image with only a small number of non-zero coefficients in the sparse representation domain [3, 4]. Herein we develop and demonstrate techniques for combining classic sparse representations with modern ideas in deep neural networks such as neural network ensembles, to improve the performance of image processing tasks such as image super-resolution. In particular, we demonstrate how our proposed techniques improve over standard baseline techniques for visual imagery, with a focus on the ability of our approach to function even in the presence of small data sets. While image processing of visual imagery is often done in the presence of large sets of training data, there are many important image processing problems that do not benefit from such large collections of training data. To that end, we also demonstrate the effectiveness of our techniques on an important problem in material manufacturing, namely the analysis of areal density maps arising from beta-particle transmission imaging of a macro-scale sheet composed of nano-scale carbon nanotubes.

Another popular technique for image-related problems is convolutional neural networks (CNNs). In the past few years, the availability of large numbers of training images have led to numerous deep neural network models, which have been developed for image denoising [5–7] and super-resolution [8, 9]. Later, a breakthrough was made by Dong et al [10] when they developed a single image super-resolution CNN (SRCNN), making CNNs practical for image super-resolution problems, and numerous studies have been developed to improve the reconstruction quality. A study of sparse priors shows that domain expertise can help to achieve improved results [11], and residual neural networks like SRResNet [12] and EDSR [13] provided improved results with the help of deep architectures.

Transform domains are widely used for signal reconstruction. In particular, the connection between low-resolution images and high-resolution images can often be clearly represented in some transform domain where, for example, the spatial redundancy in the original image is reduced in the frequency domain. The Fourier domain and higher-order wavelet sparsifying transform domains are both competitive choices for image processing problems [1]. Reducing memory requirements and computational cost compared with other types of Fourier transformations, DCT is one of the most popular selections in the Fourier family as it considers only the real part of the series [14] and many algorithms were developed in the Discrete Cosine Transform (DCT) domain before the recent breakthroughs in

image super-resolution neural networks [15–17]. Wavelet transformations have also drawn substantial interest over many years as there exists a large selection of wavelet families and bandwidths, which allows researchers to make an optimal choice based on specific image content, and wavelets have been adapted into image super-resolution neural networks to improve performance [18, 19]. One of the advantages of wavelet transformations over the DCT is that researchers can decompose the image into desired sub-bands of a given image signal. This specificity comes at a price though, as the performance of wavelets depends on how well the selected wavelet transformation suits the particular application [20], and unlike DCT there is not a "one-size-fits-all" best solution. For choosing the wavelet family, although both general guidelines [21–23] and suggestions toward specific areas [24, 25] are available, properties of different wavelets should be studied before making a selection. In real-world applications however, there are many cases where only a small collection of data is available. It can be difficult to understand the distribution of images well enough to select the appropriate wavelet *a priori* as the sample may not be representative of the entire population. Moreover, the compressive measurements are complex and diverse according to the application field. To overcome the limitation of small training data sets and the complexity of the compressive measurements, we propose an ensemble CNN for image super-resolution problems. Our new method provides a universal solution for choosing between DCT and different wavelet transformations, and gives a more robust result when compared with other advanced models in the original space domain. The neural network ensemble can learn the properties of the specific sub-image patch and matches it to the appropriate transform domain.

In this research, we demonstrate these techniques on both standard visual imagery as well as the application of reconstructing beta transmission areal density maps of MIRALON sheets. MIRALON sheets, made by Nanocomp Technologies, Inc. (Nanocomp), a wholly-owned subsidiary of Huntsman Corporation, are a non-woven material made from carbon nanotubes. Such sheets have impressive physical properties, and they provide sustainable and effective solutions to some of the toughest industry challenges involving the aerospace, energy, and electronics domains. As shown in Figure 4 (a), the seamless cylindrical hollow tube is a single-layer lattice of covalently-bonded carbon atoms [26]. These strong chemical bonds play a key role in the exceptional mechanical properties of carbon nanotube materials [27], which are further enhanced by the high length-to-diameter aspect ratio [28]. Figure 4 (b) gives a picture of the carbon nanotube network generated by an electron microscope. Formed by depositing layers directly from the furnace as they are grown, carbon nanotube sheets are not only environmentally resistant and lightweight, but also possess impressive properties including strength, thermal and electrical conductivity, high-temperature resistance, electromagnetic interference and electrostatic dissipation. The tangled carbon nanotube networks

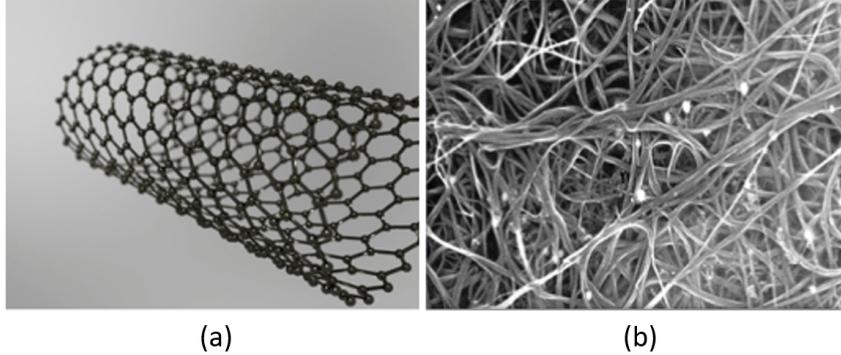


Figure 4: (a) Schematic structure of a carbon nanotube and (b) the MIRALON material under an electron microscope. The cylindrical hollow structured fiber gives MIRALON material various properties, i.e., strong, lightweight and conductive. Electrons help the long fibers stick together naturally and form a tangled network.

can be difficult to manipulate [29], and MIRALON sheets in particular, are expected to be uniform for the best performance in applications. Therefore, the quality control for a sheet with minimum areal density variation is crucial. One of the tasks in the quality control process is to generate a clear picture of areal density variation in every sheet. As shown in Figure 5, the Mahlo QMS-12 Qualiscan Beta Transmission System is designed to reproduce the general texture of a MIRALON sheet. The transmission system functions as an X-ray Camera to the sheets. As the sensing head moves over the surface, the emitter releases beta particles over an area in each picture frame. The total number of particles which reach the detector under the sheet is converted into a measurement for an areal density map of the sheet. However, the original beta transmission areal density map is insufficient to identify variation and defects at the level required in many applications. As shown in Figure 6, the distribution patterns (i.e. width and thickness of valleys and hills) and the shape and area of defects are hard to measure from the original areal density map. In this study, we propose to reconstruct a high-resolution areal density map with single image super-resolution neural networks.

However, all of the image super-resolution literature are not directly applicable in our domain for two reasons. First, the nature of the application results in a small training data task with complex camera measurement. Second, the areal density has depth dependency in the thickness direction, related to where the material is in the gap between the source and detector during the measurement. The first reason rises since there are no ground truth of high-resolution areal density maps. Using the unsupervised compressive sensing technique would require extensive knowledge of the detector on the beta transmission equipment, including how the response changes with different materials and thicknesses. However, we would like to provide a general solution for any settings of the equipment rather than re-

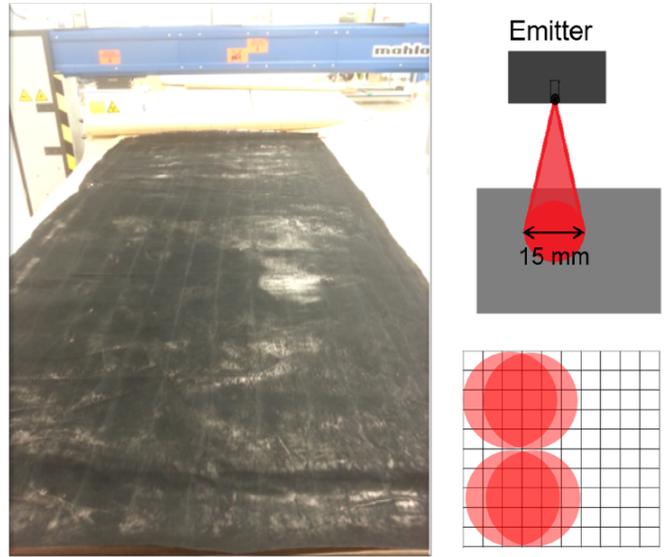


Figure 5: Mahlo QMS-12 Qualiscan Beta Transmission equipment for imaging the MIRALON sheet areal density. As the sensing head moves across the surface, the source in the upper portion releases beta particles. The number of particles which reach the detector beneath the sheet in each 20ms window is converted into a measurement.

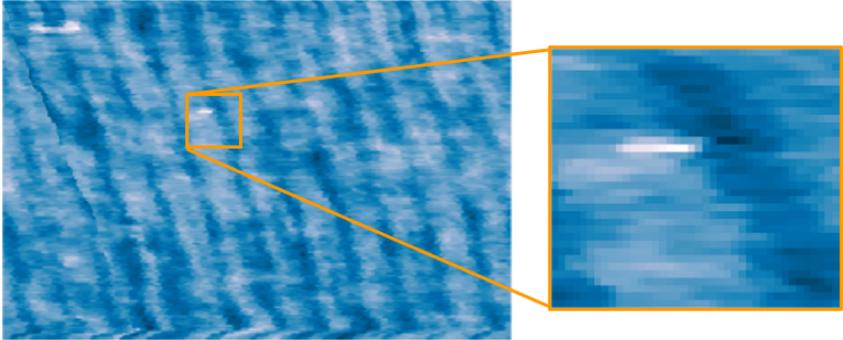


Figure 6: Areal density map of a MIRALON sheet. The transmission sensor measures the material density with a low sampling rate. The original image is insufficient to identify variation and defects at the level needed.

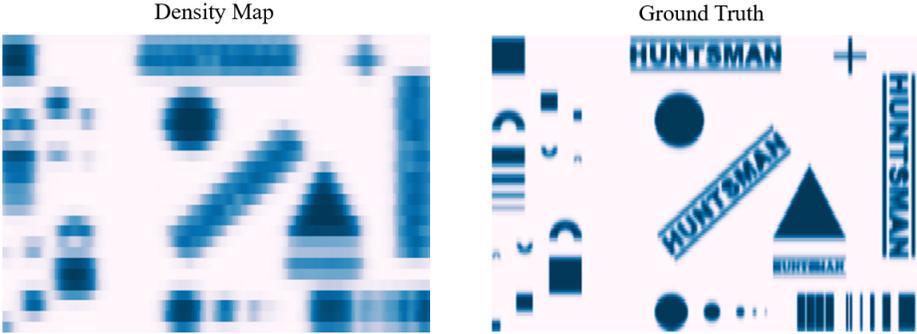


Figure 7: The designed shim-stock and its beta transmission areal density map

modeling the solution each time when changing the data collection setting. Therefore, we convert the problem into a supervised problem. We substitute a piece of shim-stock to obtain the ground truth of high-resolution areal density maps. Figure 7 shows the shim-stock and its beta transmission areal density map. An assumption of uniform areal density is made for the shim-stock. Geometric figures of different dimensions and orientations were laser-cut from the shim-stock. Then, using the same parameters used for MIRALON sheets, an areal density map was generated. This data gathering task demonstrates the need for a close collaboration between the team developing the super-resolution methodologies and the team performing the physical measurements<sup>1</sup>. Nevertheless, the amount of training data is under the requirement for training regular deep image super-resolution neural networks. We adapt the transform domain to extract more representing features from the limited information, and then use shallow architectures for more stable and efficient performances. Moreover, there are many types of transform domains that are competitive choices for image processing problems. In particular, wavelet transformations have drawn our interest since they allow researchers to make an optimal choice based on specific image content. We aim to take advantage of different transform domains for a more robust performances adapting various images with different features. Therefore, we propose ensemble CNNs, which assemble multiple sparse representatives in different transform domains for an optimal combined result. The second reason rises since the areal density is inherently a 3D phenomena. The penetrating behavior of the beta particles which are released from the transmission equipment contribute to the measurement regarding to different materials. As shown in Figure 8, the movement of the particles does not follow a linear trajectory because there are density patterns and defects in the thickness direction. Therefore, the change of density as a function of depth in the sheet is critical to estimate. Accordingly, we propose a voxel-based architecture and advanced training approach. Through a special 3-layer voxel experiment which is particularly designed for the MIRALON problem, we can gather a limited set of training data for which the true areal density is known. Given the semi-3D data, we develop a super resolution architecture and training paradigm which is simultaneously efficient in its data needs as well as being sensitive to density changes as a function of depth.

---

<sup>1</sup>Thanks to the Nanocomp team for their data generation work

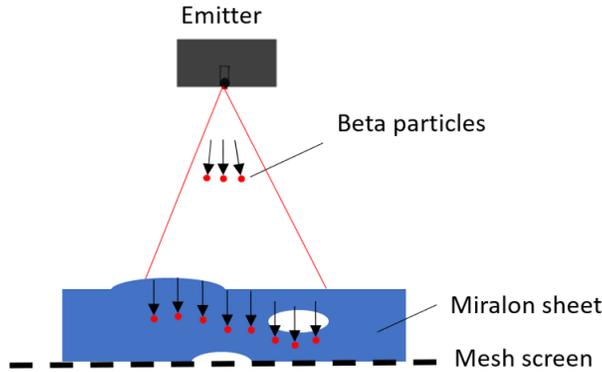


Figure 8: The camera measurement relates to the penetrating behavior of the beta particles regarding to different density patterns.

## 2 Related Works

This research develops methods for recovering high-resolution images from low-resolution images by combining ideas inspired by compressive sensing techniques with super-resolution neural networks. Compressive sensing leverages the existence of bases in which signals can be sparsely represented. Fourier and wavelet transformations are popular candidates since they provide sparse image representations. The key idea is that the frequency domain, and other transformed domains, where images can be presented sparsely can be used to ensemble a robust model and improve the performance of super-resolution convolutional neural networks. In this section we will discuss the fundamental background of compressive sensing techniques, sparsifying transformed domains and super-resolution convolutional neural networks.

### 2.1 Compressive Sensing

Compressive sensing is a signal processing technique for acquiring signals at a low sampling rate and reconstructing them to improve the resolution. It provides an efficient alternative to traditional methods which require the acquisition of data with a high sampling rate. Classically the Nyquist-Shannon sampling theorem states that a signal  $y_t \in R^N$  can be reconstructed by  $M (M < N)$  linear measurements if the sampling rate is more than twice the highest frequency. As expressed in equation (1), if the original signal  $y_t$  is sparse in a certain basis  $\Psi$  in which the restricted isometry property or similar assumptions hold, then it can be reconstructed by the compressed signal  $y_s$  and the sensing matrix  $R$  [4] [30]. For example, choosing  $\Psi$  as the Fourier basis leads to many natural signals being sparse in that domain. For  $R$ , the choice is crucial and it has been shown that Bernoulli and Gaussian random matrices are good candidates [4] [30]. Based on these assumptions, standard approaches use

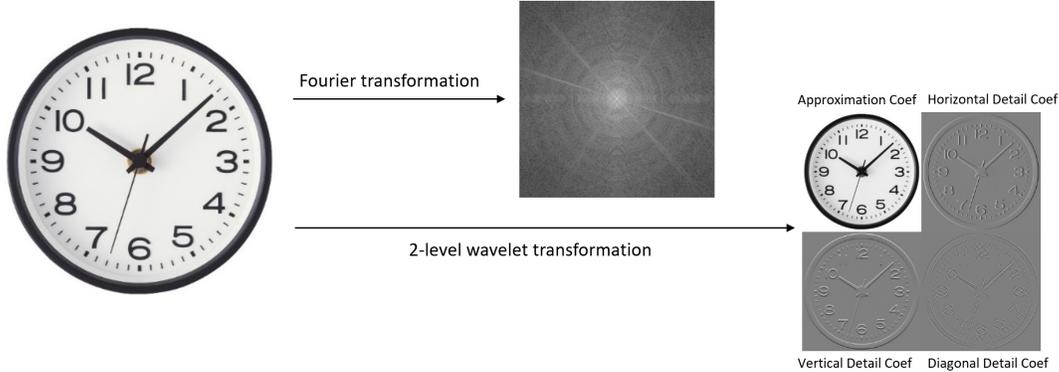


Figure 9: Transformations separate frequencies in different ways. Sparse representatives can deliver the information more efficiently with no loss to help with the small training data challenge.

$l_1$  regularization and the desired sparse solution can be obtained by solving the optimization problem

$$\begin{aligned} \min_{\omega \in R^M} \quad & \|\omega\|_1 \\ \text{s.t.} \quad & R\Psi\omega = y_s. \end{aligned} \tag{1}$$

With the optimal solution  $\omega^*$ , the original signal can be reconstructed as  $y_t = \Psi\omega^*$ . Classical algorithms for compressive sensing solve the problem by iteratively reweighting the  $l_1$  minimization [31] [32] [33] or other similar techniques. Improvements can be made by studying the signal sparsity from prior knowledge of the image [34] [35] [36].

Reflecting on the success of compressive sensing methods, we are inspired to extend current CNN based super-resolution techniques by leveraging appropriate sparse representations.

## 2.2 Transform Domains

Transform domains are widely used for signal reconstruction. In particular, the connection between low-resolution images and high-resolution images can often be represented in some transform domain where, for example, the spatial redundancy in the original image is reduced in the frequency domain. The Fourier domain and higher-order wavelet sparsifying transform domains are both competitive choices for image processing problems [1]. In particular, we focus on DCT and several 2-level wavelet transformations in this study. As shown in Figure 9, transformations separate frequencies in different ways. Sparse representatives can deliver the information more efficiently with no loss to help with the small training data challenge.

Fourier transforms decompose signals into frequencies. The Fourier transform of a function is a complex-valued function representing the complex sinusoids that comprise the orig-

inal function. The general definition for the Fourier transformation of a function  $f(x)$  is

$$\hat{f}(\xi) = \int_{-\infty}^{\infty} f(x)e^{-i2\pi\xi x} dx, \forall \xi \in R, \quad (2)$$

stating that the transformation of  $f(x)$  at frequency  $\xi$  is a complex number  $\hat{f}(\xi)$ . Writing a signal in terms of a complex exponential (frequency component) not only helps to understand the signal characteristics but also makes it easier to manipulate [37]. However, the infinite integral cannot be applied directly in computational algorithms. To bring Fourier transformation into real-world engineering applications, discrete Fourier transform (DFT) was introduced by Shmuel Winograd in 1976 [38]. DFT deals with a finite amount of data by converting a finite sequence of equally spaced samples into a same-length sequence. A sequence of  $N$  complex numbers  $\{x_n\} = x_0, x_1, \dots, x_{N-1}$  can be converted into another sequence of complex numbers  $\{X_n\} = X_0, X_1, \dots, X_{N-1}$  by

$$X_k = \sum_{n=0}^{N-1} x_n \cdot [\cos(\frac{2\pi}{N}kn) - i \cdot \sin(\frac{2\pi}{N}kn)]. \quad (3)$$

DFT has been widely implemented by fast Fourier transform (FFT) algorithms in digital signal processing problems. When applied, the summation term is separated into a real part and a complex part that need to be computed individually. Proposed in 1972 by Nasir Ahmed [39], DCT is considered the most efficient encoder for image compression. It is capable of compressing image data size by 8 times compared to the original signal. In 1995, the lossless version of DCT was developed, making it a more efficient image compression algorithm than entropy coding [40]. Reducing memory requirements and computational cost compared with other types of Fourier transformations, DCT is one of the most popular selections in the Fourier family for our research as it considers only the real part of the series [14] and many algorithms were developed in the DCT domain before the recent breakthroughs in image super-resolution neural networks [15–17]. Instead of using a summation of complex numbers in DFT, DCT uses a sum of cosine functions. A finite sequence of data points can be transformed into another sequence of real numbers oscillating at different frequencies. There are several variants of the DCT definitions, and we use the most common Type II DCT in this research. In particular, a sequence of  $N$  real numbers  $x_0, x_1, \dots, x_{N-1}$  can be converted into another sequence of real numbers  $X_0, X_1, \dots, X_{N-1}$  by

$$X_k = \sum_{n=0}^{N-1} x_n \cdot \cos[\frac{\pi}{N}(n + \frac{1}{2})k]. \quad (4)$$

Alternatively, DCT can be interpreted as taking the DFT of a sequence  $y$  of length  $2N$  which is the concatenation of the original sequence with its reverse. Since  $y = x_0, \dots, x_{N-1}, x_{N-1}, \dots, x_0$  is symmetric, its DFT contains only the real part.

Wavelet transformations have also drawn substantial interest over the last several years as a large selection of wavelet families and bandwidths exist, which allows researchers to select the optimal choice based on specific image content. Wavelets have been adapted into image super-resolution neural networks to improve performance [18, 19]. "Wavelet" in wavelet transformation can be understood as "oscillation". Beginning at zero, a wavelet is a wave-like oscillation with an amplitude that returns to zero one or more times. By a certain orthonormal series generated from a wavelet, we can obtain a wavelet series as a square-integrable function. In a square-integrable function, the integral of the square of the absolute value is finite. Function  $\psi(x)$  is square-integrable on interval  $[a, b]$  if

$$\int_a^b |\psi(x)|^2 dx < \infty. \quad (5)$$

$\psi(x)$  can be called an orthonormal wavelet if it can be used to define a Hilbert basis  $\{\psi_{jk} : j, k \in Z\}$ . Then  $f(x)$  is a wavelet series if it can be expanded in this basis as

$$f(x) = \sum_{j,k=-\infty}^{\infty} c_{jk} \psi_{jk}(x), \quad (6)$$

where  $c_{jk}$  are the wavelet coefficients.

The fundamental idea of using wavelet series is that by choosing suitable basis functions, it allows only changes in time extension while the shape remains the same. Researchers can choose the analysis frequency of the basis function by changing the time extension. In other words, one of the advantages of wavelet transformations over the DCT is that researchers can decompose the image into desired sub-bands of a given image signal. At specific frequencies, wavelet transformations are less sensitive and have lower computational cost [1]. This may help with our small training data challenge because we require a more robust algorithm that is less sensitive as our training sample may not be representative of the entire population. However, the performance of wavelets depends on how well the selected wavelet transformation suits the particular application [20]. For choosing the wavelet family, although both general guidelines [21–23] and suggestions toward specific areas [24, 25] are available, properties of different wavelets should be studied before making a selection. For example, Figure 10 and Figure 11 show some wavelet transformations of  $f(x) = \sin(x)$ . In this research, Daubechies [41], biorthogonal, and reverse biorthogonal for wavelet domains [23, 42, 43] are

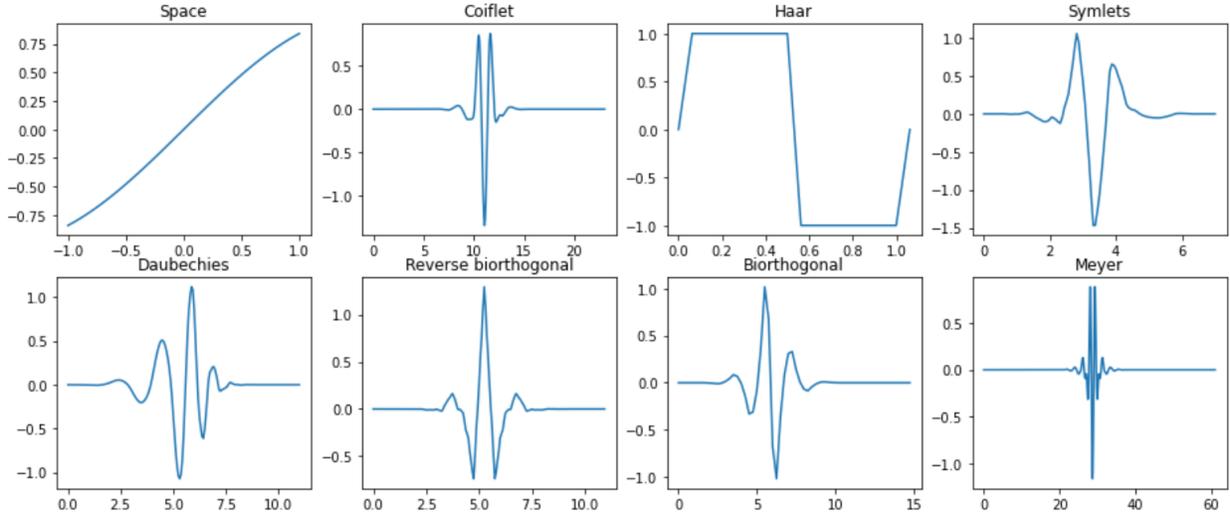


Figure 10: Examples of different wavelet transformation of  $f(x) = \sin(x)$ . A large selection of wavelet families exists. Different wavelets transform the signal into different frequencies.

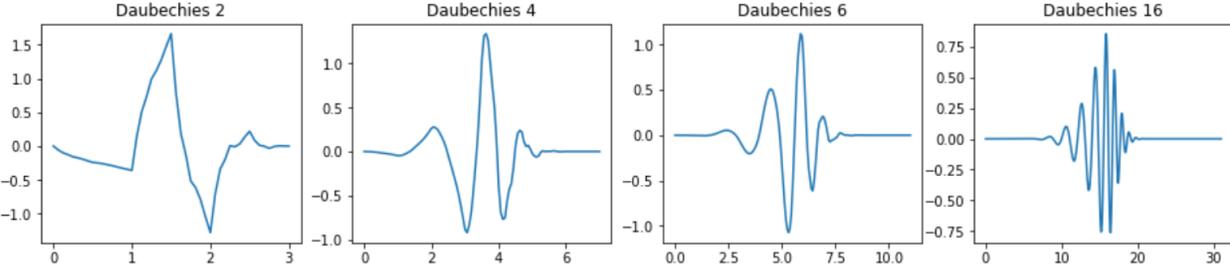


Figure 11: Examples of different bandwidths of Daubechies wavelet transformation of  $f(x) = \sin(x)$ . Different bandwidths within a wavelet family exist.

considered. Figure 12 shows 2D wavelet representatives of a 10x10 pixel shim-stock patch from different wavelet families. As shown in the example, there are multiple bandwidth to choose from within each family. Figure 13 shows an example of first level wavelet transformations of an ordinary image. Daubechies 6 and biorthogonal 3.7 transformations separate frequencies into detail coefficient matrices, providing frequency details differently.

In our study, we use multi-channel wavelet representations specifically. As illustrated in Figure 14, a two-level wavelet transformation separates the horizontal, the vertical and the diagonal details from the approximation. The entire image matrix can be divided into four detail sub-matrices. We hope to avoid extracting features from the intersection of different sub-matrices in order to improve the efficiency. The four detail sub-matrices are fold into a 4-channel representative. In this way, each feature in our model is extracted from only one sub-matrix.

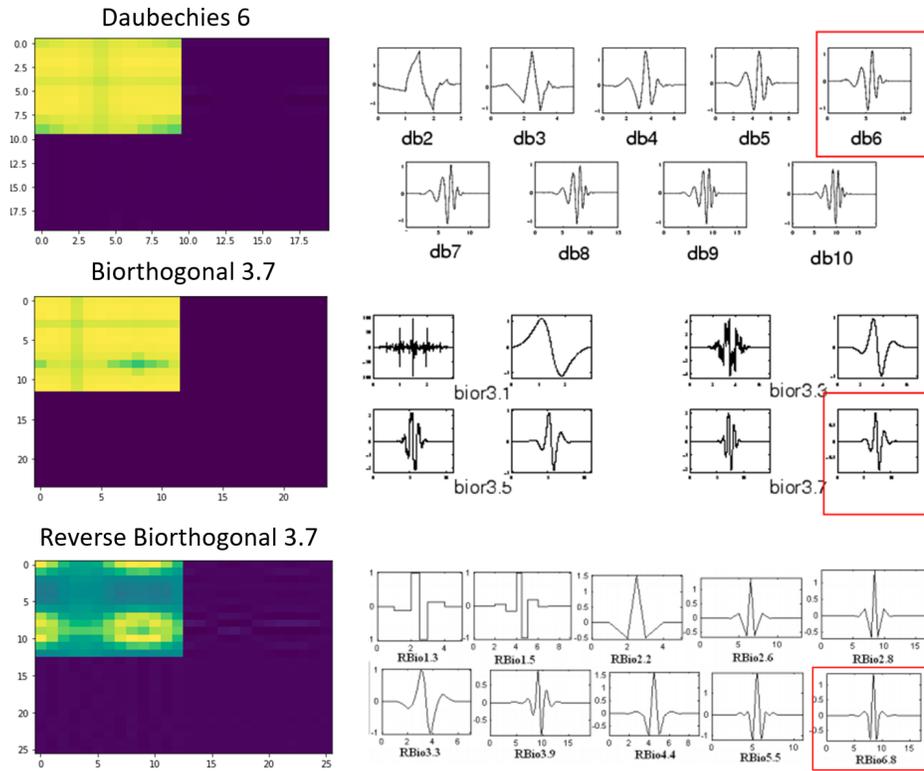


Figure 12: 2D wavelet representatives of a 10x10 pixel patch from different wavelet families. There are multiple bandwidth to choose from within each family.

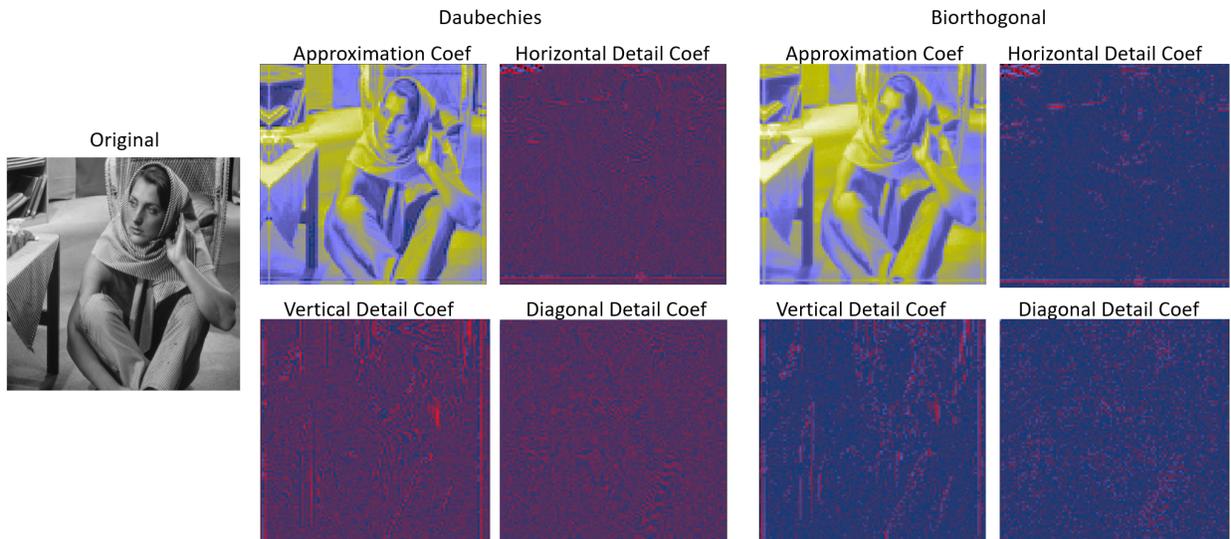


Figure 13: Wavelet detail coefficient matrices from first level Daubechies 6 and biorthogonal 3.7 transformations.

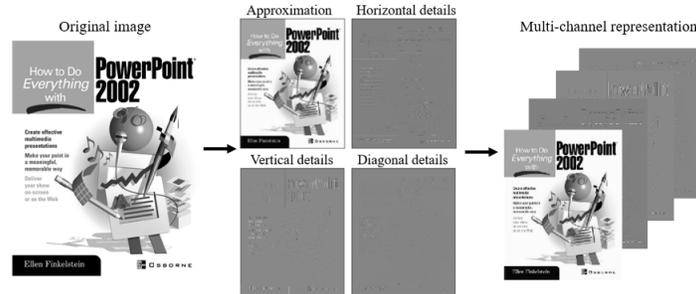


Figure 14: Reprocessing wavelet representatives. A 2-level wavelet transformation gives four detail sub-matrix. The multi-channel representative is obtained by folding the detail sub-matrix.

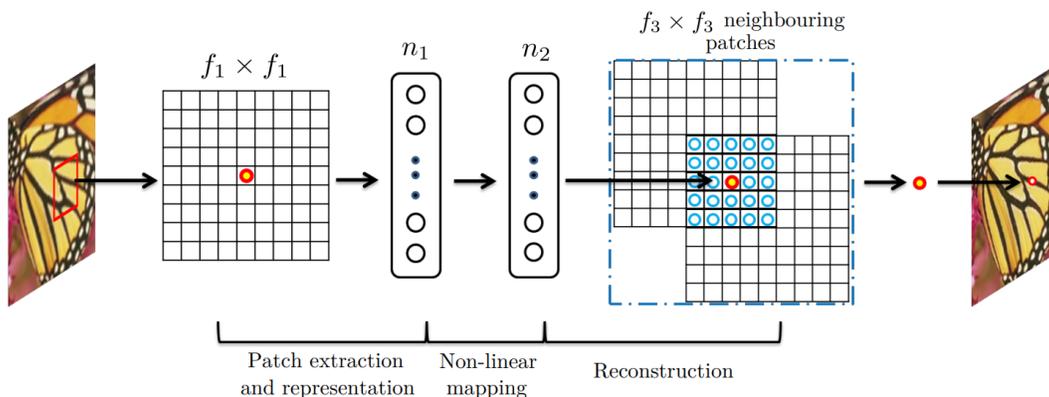


Figure 15: An illustration of sparse-coding-based methods in the view of a convolutional neural network in SRCNN [10]. Each pixel in the high-resolution image can be reconstructed from a neighborhood of several pixels in the low-resolution image.

### 2.3 Super-resolution CNNs

CNNs have become a very popular technique for image processing problems. They were initially applied to image classification problems [44] and achieved success in many fields such as facial recognition [45] and object detection [46]. Subsequently, residual learning and auto-encoders were applied to studies of image denoising [6] [7]. Based on previous work, Chao et al [10] proposed a single image super-resolution neural network (SRCNN), providing an end-to-end mapping from low-resolution patch inputs to high-resolution pixel output. It is demonstrated that the sparse-coding based algorithm, which is mathematically similar to compressive sensing and can be used for finding the sparse base  $\Psi$  in Equation (1), can be viewed as a convolutional neural network. As shown in Figure 15, SRCNN uses the neural network as an optimized pipeline consisting of three operations; patch extraction and representation; non-linear mapping in high dimensional space; and reconstruction of high dimensional representations. Our model is inspired by, and benefits from, SRCNN.

Following the breakthrough made by single image super-resolution neural networks, various authors have worked on adapting compressive sensing techniques to CNNs. Wang et al [47] combine sparse coding with a CNN to improve the performance of the image super-resolution problem. In this case the under-determined system is

$$y_t = R^{-1}y_s + e \tag{7}$$

where  $e$  is the measurement noise. With respect to an over-complete dictionary  $D$  (equivalent to the sparse base  $\Psi$  in Equation (1)), the image can be represented by sparse 'linear' coefficients  $\alpha$ . With a well-defined dictionary,  $y_t$  can be recovered by solving Equation (8), where  $\lambda$  is the regularization coefficient. However, since the convolution operation (with activation functions) is nonlinear with respect to the entire image, the recovered sensing measurement  $R$  is not necessarily linear.

$$\begin{aligned} y_s &= D\alpha \\ \text{s.t. } \alpha &= \underset{\alpha}{\operatorname{argmin}} \|y_t - R^{-1}D\alpha\|_2^2 + \lambda\|\alpha\|_1. \end{aligned} \tag{8}$$

Subsequently, a simpler scalable single image super-resolution CNN was introduced by Kulkarni et al [48]. Here a deep neural network is used in place of the traditional  $l_1$ -minimization method. They state that the image signal is not exactly sparse with respect to the dictionary  $D$ . A convolutional architecture which refines the image estimation in each iteration is efficient enough to estimate the sensing measurement  $R$ . From this work, Kulkarni et al developed a method called ReconNet. A significant benefit of ReconNet is better performance with few parameters, somewhat reducing the computational cost. As shown in Figure 16, a linear layer is used to up-scale the image patches and then fully connected with convolutional layers for image reconstruction. This can also be interpreted as ReconNet using both linear layers and nonlinear layers to recover the sparse base  $\Psi$  and the sensing measurement  $R$  simultaneously. We propose to improve this model using frequency domains. With the help of sparse representations in the frequency domain, a shallower architecture with lower computational cost can result in better performance.

The development of single image super-resolution neural networks also progressed in a different direction with residual neural networks. Residual neural networks use deep residual block modules and skip connections to explore the feature space in depth to avoid vanishing gradients. Numerous models have been developed increasing the accuracy of the reconstruction i.e., SRResNet [12], EDSR [13], RCAN [49]. EDSR improves the deep residual blocks in SRResNet by removing the batch normalization layers since they hurt the flexibility when normalizing the features. RCAN uses channel-attention residual blocks to increase

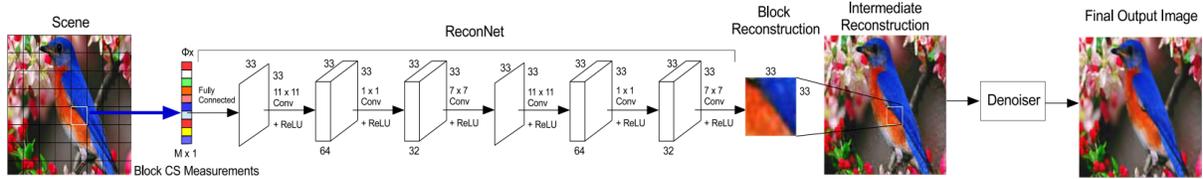


Figure 16: The architecture of ReconNet [48]. A linear layer up-scales the image patches and then fully connect to 6 convolutional layers with non-linear activation function for image reconstruction.

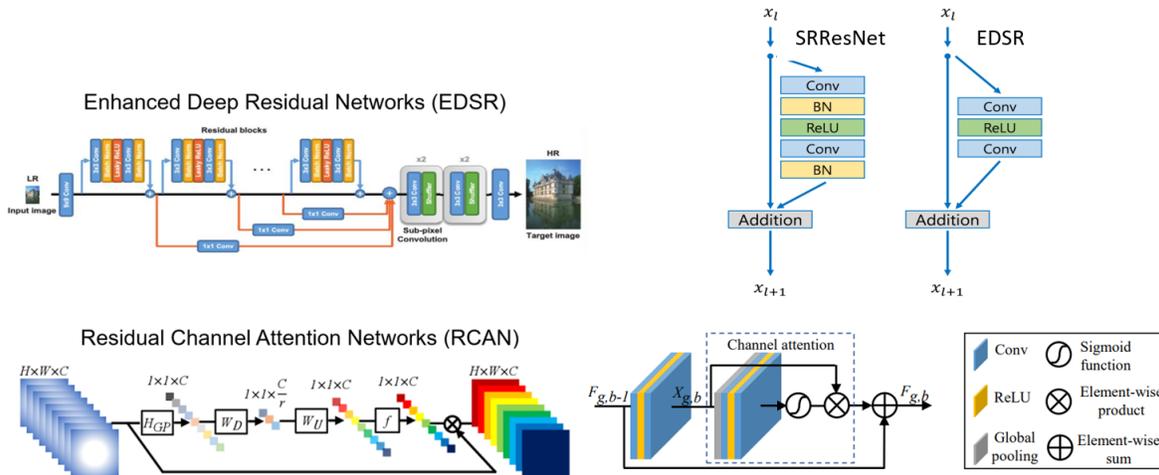


Figure 17: The architectures of EDSR [13] and RCAN [49] with corresponding featured residual blocks. EDSR improves the deep residual blocks in SRResNet by removing the batch normalization layers. RCAN uses channel-attention residual blocks to reduce the impact from abundant low-frequency information and increase the network’s representational ability.

the network’s representational ability. Figure 17 shows the general architectures of EDSR and RCAN with their featured residual blocks. Although these models may not be suitable for our problem since the gradient decent in deep architecture cannot be optimized on small training data, we use these high-performance benchmark models as baselines to make comparison in the experiments.

### 3 Methodology for Data Science

In contrast to the ideas from Kulkarni et al for ReconNet [48], we propose a different explanation for scalable single image super-resolution CNNs. The reason that the convolutional architecture works well for the under-determined linear system is not in contrast to the pre-defined sparsity assumption in the traditional method. The model is learning the sparsity and the sensing measurement simultaneously by training parameters in  $R$  from equation (3). In addition, we do have knowledge that common image signals are sparse in the Fourier domain, and the high-resolution image can accurately be recovered from the low-resolution image by their sparse representations [50]. Accordingly, we believe that the sparsity of our target images in the frequency domain leads to better reconstruction for high-resolution images, *even when the neural network is trained on limited low-resolution frequency information*. Furthermore, we challenge the deep architecture used in previous studies. Numerous convolutional layers suffer from exploding/vanishing gradients [51]. Unlike previous studies on deep residual-learning and gradient clipping [52] [53], we propose to reduce the computational cost and increase the efficiency by a shallower architecture in the frequency domain.

Our model outperforms traditional compressive sensing methods by learning a non-linear transformation instead of defining a linear sensing matrix  $R$ . It also improves upon previous scalable single image super-resolution CNNs by learning the measurements in the sparsifying transform domains. Previous studies used a fully connected layer between low dimensional space domain and high dimensional feature domain. We propose to learn features from the transform domain in order to increase the computational efficiency. The representation in the transform domain can be easily found by

$$x = Fy_s, \quad y = Fy_t \tag{9}$$

where  $F$  denotes the operation of the sparsifying transformation. Training the model for

$$y = R_f^{-1}x + e_f \tag{10}$$

can help to improve the performance. Here,  $e_f$  is the measurement noise.

The Fourier domain and higher-order wavelet sparsifying transform domains are competitive choices for image processing problems [1]. The researchers can choose desired transform domains based on the image content and the particular application field. However, it is hard to determine the best suited transform domain when the amount of training data is limited. In our study, we focus on image super-resolution problems with small training data. The available training data cannot be assumed to be representative of the entire population. In

addition, as we have a particular interest in the beta transmission images for MIRALON sheets, the compressive measurements are more complex than the cases for ordinary images. Therefore, we propose an ensemble method with multiple transform domains to overcome these challenges. Various sparsifying representatives from multiple domains are provided to the neural network at the same time. In other words, we manually transform the signal into different frequencies and directions to enrich the information so that the neural network can choose the desired representatives automatically and process the signal more efficiently. Furthermore, the neural network ensemble can learn the properties of the full image and combines the results from different transform domains based on specific image content, delivering a more robust algorithm.

In the following sections, we will introduce 5 models in the order of our developments, illustrating the advantages of (1) sparsifying transform domains, (2) shallow architectures, and (3) ensemble methods for solving image super-resolution problems with complex compressive measurements where only a small collection of training data are available. The study is conducted with five steps:

- First, we developed **SparseFnet** to prove that image representations in the frequency domain can be more effectively processed by CNNs. It also shows that for the challenge of small training data, a shallower architecture with sparse representations can provide a more stable and efficient result or reconstruction.
- Second, we included more choices of sparsifying transformations for the ensemble to consider, generating a more robust algorithm for providing a general solution to small data image super-resolution problems for different image content in different application fields. In addition to the Fourier transformation, multiple wavelet transformations are considered by **ESnet** to improve the performance over single domain SparseFnet.
- Third, we developed **EWnet** as a more uniform version of ESnet. The best-suited sparsifying transformations are determined for different sub-image-patches and different compressive measurements at the same time as the training of Super-resolution CNNs.
- Next, we further expanded the ensemble method. We propose a model called **Ensem-Net** to study the performance of combining different bench mark models in transform domains to prove that the ensemble helps to improve the performance.
- Finally, we extended the study to a semi-3D model. We proposed a **Voxel-design** with specific consideration of the beta transmission image applications. This considers

the depth dependency of the compressive measurements. By introducing a 3-layer template design, we successfully extended the previous results to 3D space.

### 3.1 The comparison between architectures of different algorithms

This section is a summary of the comparisons between our proposed algorithms. Figures 18, 19, 20 and 21 are simplified architectures of the algorithms highlighting the differences and the key todos/features. The detailed architectures will be illustrated in the individual sections for each algorithm. We did not include the figure for the last voxel-based design since it is an application-driven modification of EnsemNet. There are several essential features in this study for the challenges of small training data and complex compressive measurements. Table 1 lists all the features across the different algorithms.

SparseFnet is an improved scalable single image super-resolution neural network in the frequency domain. Inspired by the sparsity assumption in the Fourier domain from standard compressive sensing methods, a new architecture is built. A primary feature of our model is that the neural network uses **Fourier loss**. Illustrated in Figure 18, the neural network is trained with respect to the difference between Fourier representatives rather than the difference between the images in the space domain. Moreover, we simplify the architecture of deep neural networks for image reconstruction problems. The **shallower architecture** is demonstrated to be just as accurate with lower computational cost. Our experimental results illustrate that the proposed SparseFnet significantly outperforms previous scalable single image super-resolution neural networks.

Table 1: Essential features of the different algorithms developed in this work

| Approaches                      | Models     |       |       |          |
|---------------------------------|------------|-------|-------|----------|
|                                 | SparseFnet | ESnet | EWnet | EnsemNet |
| Fourier loss function           | ✓          |       |       |          |
| Shallow architecture            | ✓          | ✓     | ✓     | ✓        |
| Multiple sparsifying transforms |            | ✓     | ✓     | ✓        |
| Ensemble method                 |            | ✓     |       | ✓        |
| Input sparse representatives    | ✓          | ✓     | ✓     | ✓        |
| Output sparse representatives   | ✓          |       | ✓     |          |

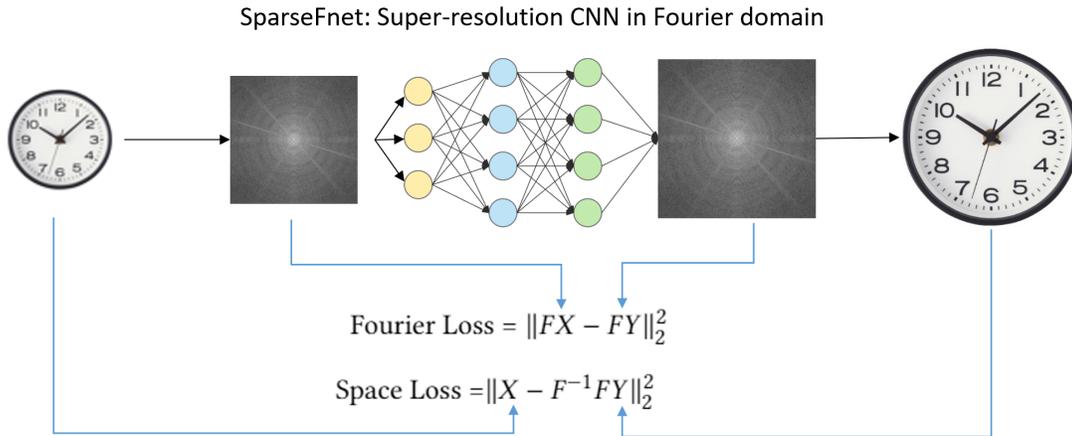


Figure 18: SparseFnet. Key features: (1) shallow architecture, (2) the model uses Fourier loss. Input: sparse representative (Fourier). Output: high-resolution sparse representative (Fourier).

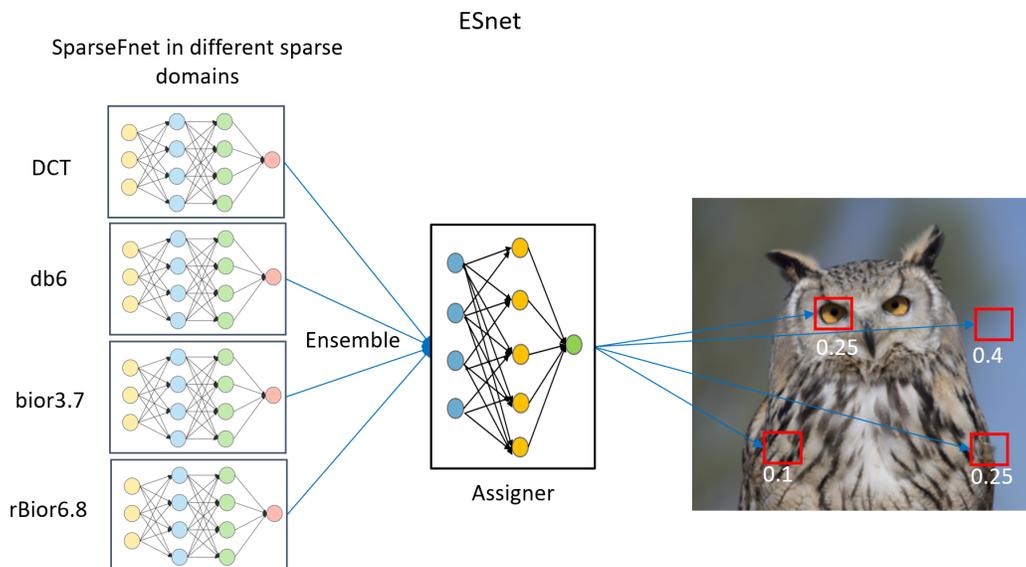


Figure 19: ESnet. Key feature: ensemble reconstruction from different SparseFnet results which are determined in different domains. Input: the original low-resolution image and multiple sparse representatives (Fourier and wavelets). Output: high-resolution reconstruction of the image.

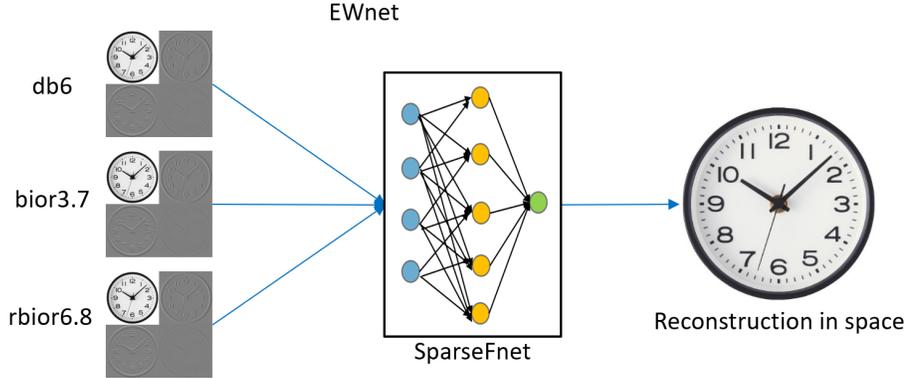


Figure 20: EWnet. Key feature: more robust extraction of features across different transform domains than ESnet. Input: multiple sparse representatives (wavelets). Output: high-resolution reconstruction of the image.

ESnet is an ensemble algorithm for selecting the appropriate sparse domain for the image super-resolution problem. Instead of choosing the best-suited sparsifying transformation for the entire image, our proposed ESnet leverages the advantages from **multiple transform domains across different sub-image patches**. As shown in Figure 19, SparseFnets reconstructions from several sparsifying domains are generated first. Then, an assigner neural network assigns weights to the reconstruction patches from the different domains. The combined result is a universal solution for various types of images. The advantages of using ESnet are illustrated through experiments on both regular images and the areal density maps of MIRALON sheets for different complex compressive measurements. Specifically, ESnet provides a more robust solution for the limitation of small training data sets.

Similar to ESnet, EWnet leverages the advantages from multiple transform domains across different sub-image patches but works in a more concise fashion. Figure 20 gives the simplified architecture. Sparse representatives from multiple domains are fed into a shallow SparseFnet. Unlike the basic SparseFnet that outputs high-resolution Fourier representatives, EWnet directly outputs the high-resolution reconstruction of the image. As such, EWnet provides a more robust and efficient solution even with the limitation of small training data sets. The advantages of using EWnet are illustrated through experiments on both regular images and the areal density maps of MIRALON sheets for different complex compressive measurements. We reveal additional texture of the carbon nanotube sheets by applying EWnet to MIRALON areal density maps in the next section.

EnsemNet is another improved version of ESnet. EWnet improves the robust performance by leveraging the advantages from sparsifying features across multiple transform

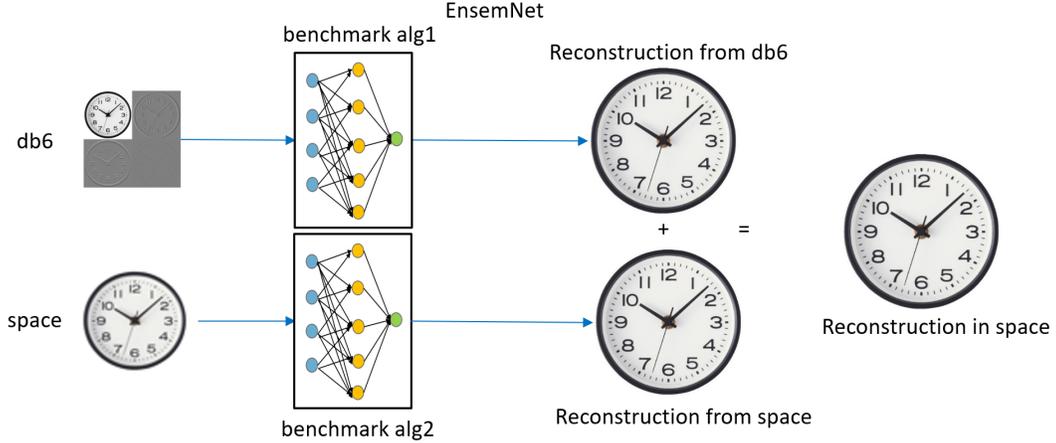


Figure 21: EnsemNet. Key feature: improved version of ESnet focusing on ensemble, the initial algorithm for the voxel-based semi-3D method. Input: the original low-resolution image and the sparse representative (wavelets). Output: high-resolution reconstruction of the image.

domains, while EnsemNet increase the robust performance by an ensemble from different algorithms. A more general solution is provided for various types of applications by combining the output from different benchmark models which are performed on transform domains. As illustrated in Figure 21, two shallow versions of benchmark neural networks in different transport domains are performed, and the output is an element-wise summation of the two reconstructions. EnsemNet provides a more robust and efficient solution for the image super-resolution problem with small training data sets and diverse compressive models.

Finally, in order to identify variation and defects in the areal density maps of MIRALON sheet at the required level, we need sufficient accuracy in the thickness direction. Current 2D super-resolution neural networks cannot provide enough information. Therefore, we propose a **3-layer voxel representation** to adapt previous ensemble CNNs in transform domains to the particular application of MIRALON sheets. EnsemNet is adapted into this method. Using our proposed techniques, a high-resolution areal density map, which is rich in 3D information, can be obtained by leveraging 2D image super-resolution CNNs.

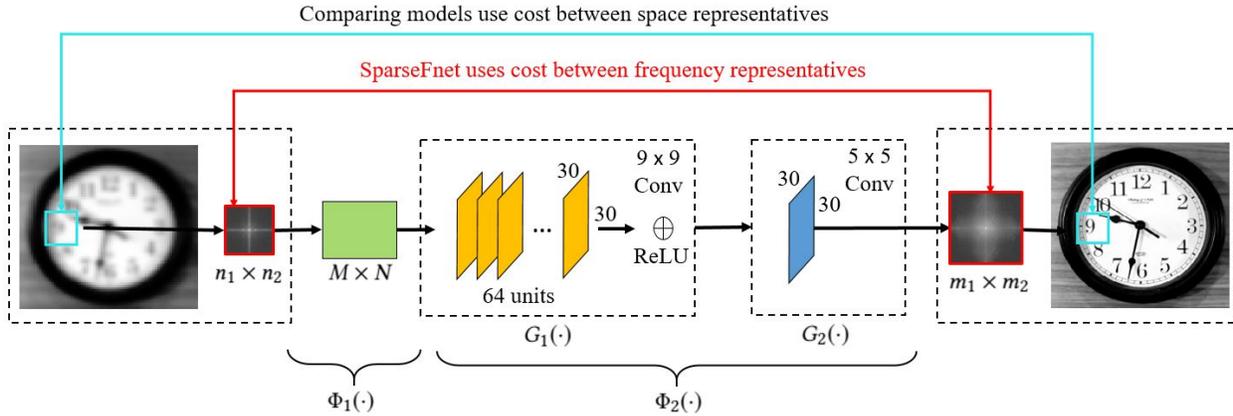


Figure 22: The shallow architecture of our proposed models. The first linear layer maps input patches to high dimensional feature space. The second convolutional layer extracts features non-linearly. The third convolutional layer acts as an averaging filter of features and reconstructs high dimensional patches.

### 3.2 SparseFnet: Shallow Super-resolution CNN in the Frequency domain

Taking advantage of signal sparsity, we developed a model in the frequency domain with shallow architecture called SparseFnet. It improves the performance of previous image super-resolution neural networks. Performing the operation on a small number of non-zero coefficients in the frequency domain, SparseFnet provides a more stable model that effectively increases the accuracy and lowers the computational cost. Herein, we introduce three proposed models, shallow neural network (Shallownet), sparse domain neural network in the frequency domain with spatial cost (SparseSnet), and sparse domain neural network in the frequency domain with frequency cost (SparseFnet). We use similar architectures for these three models in different domains to demonstrate the advantage of SparseFnet.

#### The architecture of SparseFnet

The framework of our models is shown in Figure 22. Before training the model, we preprocess the data by extracting overlapped patches from the entire image. The model is then trained on these small image patches. The framework contains a linear layer  $\Phi_1(\cdot)$  and convolutional layers  $\Phi_2(\cdot)$ . The linear layer takes small patches from the low dimensional space as inputs and feeds preliminary reconstructed patches on the high dimensional space to the convolutional layers. Then the convolutional layers apply nonlinear filters to further improve the reconstruction accuracy.

The first layer in the neural network is a fully connected linear layer. A compressed signal  $x \in R^{m_1 \times n_2}$  ( $N = n_1 \times n_2$ ) is projected to the high dimensional space by a linear mapping function  $\Phi_1(\cdot)$ . A preliminary reconstructed signal  $y_1 \in R^{m_1 \times m_2}$  ( $M = m_1 \times m_2$ ) is obtained by

$$\begin{aligned} y_1 &= \Phi_1(x) \\ \Phi_1(x) &= W_1 \cdot x + b_1 \end{aligned} \tag{11}$$

where  $W_1$  is a  $M \times N$  dimensional linear filter and  $b_1$  is a  $M$  dimensional bias vector. Note that  $\Phi_1(\cdot)$  is not necessarily representative of the sensing measurement. It only represents projections between different domains in the neural network.

The convolutional part of our architecture,  $\Phi_2(\cdot)$ , has two convolution layers,  $G_1(\cdot)$  and  $G_2(\cdot)$ , which are both in the projected high dimensional space. These two layers are used for feature extraction and patch reconstruction respectively. The first layer  $G_1(\cdot)$  extracts  $k_1$  features from the preliminary reconstructed signal  $y_1$  by non-linear filters, and ReLU is chosen as the activation function. The layer is expressed as

$$\begin{aligned} y_2 &= G_1(y_1) \\ G_1(y_1) &= \max(0, W_2 \cdot y_1 + b_2) \end{aligned} \tag{12}$$

where  $W_2$  contains  $k_1$  filters of size  $g_1 \times g_1$  and  $b_2$  is a  $k_1 \times g_1 \times g_1$  dimensional bias vector. Then  $G_2(\cdot)$  acts as an averaging filter for patch reconstruction. The output of the second non-linear layer is the final reconstructed patch  $y \in R^{m_1 \times m_2}$ . Each pixel in  $y$  can be viewed as a weighted average of the  $k_1$  features from the previous layer  $G_1(\cdot)$ . We define this layer by

$$\begin{aligned} y &= G_2(y_2) \\ G_2(y_2) &= W_3 \cdot y_2 + b_3 \end{aligned} \tag{13}$$

where  $W_3$  contains  $k_2$  filters of size  $g_2 \times g_2$ , and  $b_3$  is the corresponding bias vector of size  $m_1 \times m_2$ . The settings for these parameters in our experiments are  $k_1 = 64, g_1 = 9, k_2 = 1, g_2 = 5$ .

The fundamental structure of our models consist of the previously introduced linear and convolutional layers. We adapt this structure to construct ShallowNet, SparseSnet and SparseFnet with different settings.

ShallowNet is the baseline of our proposed models. It can be viewed as a simplified version of ReconNet using the framework shown in Figure 22. The low-resolution patch is processed by the linear layer and two convolutional layers sequentially, then a high-resolution patch is output for reconstructing the whole image. We do not apply additional operations other than the operations illustrated in Figure 22. The input signal  $x$  and the output  $y$

are both image patches in the space domain. Suppose the operation of the network can be written as  $\Phi_\theta = \Phi_2(\Phi_1(\cdot)) \in R^{M \times N}$  where  $\theta$  denotes the parameters in the network. The high-resolution patch can be recovered by

$$\begin{aligned} y &= y_t, x = y_s \\ y &= \Phi_\theta(x) \\ \min_{\theta} \|A - \Phi_\theta(x)\|_2^2 \end{aligned} \tag{14}$$

where  $y_s$  and  $y_t$  are the compressed patches and the reconstructed patches in the space domain respectively and  $A$  is the ground truth for the high-resolution patches. In the case where the neural network is equivalent to regular compressive sensing approaches,  $\Phi_\theta$  for Shalownet can be expressed as a sparse sensing matrix. In other words, the network learns the projection for the sparsity domain and the sensing measurement simultaneously. If the sparsity projection is given, then the algorithm can be more efficient with the same architecture. Thus, we propose SparseSnet and SparseFnet in the frequency domain.

As described above, the Fourier transformation is a good candidate for the sparsity domain. For both SparseSnet and SparseFnet, the input and the output patches are pre-processed by the Discrete Cosine Transformation (DCT) operation  $F$ . SparseSnet performs the optimization in the space domain, while SparseFnet optimizes in the frequency domain directly.

The optimization of SparseSnet is

$$\begin{aligned} y &= Fy_t, x = Fy_s \\ y &= \Phi_\theta(x) \\ \min_{\theta} \|A - F^{-1}\Phi_\theta(x)\|_2^2. \end{aligned} \tag{15}$$

The network uses the inverse DCT on high dimensional Fourier representative patches to compute the cost. SparseSnet is in the frequency domain but uses gradient descent with respect to the cost in the space domain.

SparseFnet is more straight forward in the frequency domain. The network learns  $\Phi_\theta$  from

$$\begin{aligned} y &= Fy_t, x = Fy_s, \Omega = FA \\ y &= \Phi_\theta(x) \\ \min_{\theta} \|\Omega - \Phi_\theta(x)\|_2^2. \end{aligned} \tag{16}$$

$\Omega$  is the Fourier representation of the ground truth  $A$ . Here the cost in each iteration is computed between Fourier representations directly. An estimate of  $\Omega$  is output in the fre-

---

**Algorithm 1** SparseFnet

---

**Input:**  $y_s \in R^{n_1 \times n_2}$ ,  $A \in R^{m_1 \times m_2}$ Pre-training process:  $x = Fy_s$ ,  $\Omega = FA$ .Initialize  $\Phi_\theta$  with randomly initialized parameters.**While** epoch = True  **for** each iteration     $y = \Phi_\theta(x)$     Minimize  $\|\Omega - \Phi_\theta(x)\|_2^2$  using back-propagation.  **end for****end while****Return**  $y$ Post-training process:  $y_t = F^{-1}y$ .**Output:**  $y_t \in R^{m_1 \times m_2}$ 

---

quency domain after which the inverse DCT is applied to the estimate in order to recover the representation in the space domain. Unlike SparseSnet, SparseFnet never sees the space domain. This network is built on the assumption that the DCT projection provides an alternative basis on which the signal is approximately sparse. The projection for the sparsity domain is found during pre-training, and the operation  $\Phi_\theta$  learns only the sensing measurement in the frequency domain. If the assumption of sparsity stands, then SparseFnet should perform better than Shallownet and SparseSnet. The training pseudo-code for SparseFnet is provided in Algorithm 1.

### SparseFnet on ordinary image reconstruction

Several experiments were conducted on regular images to demonstrate the effectiveness of our proposed techniques. The labels for training the neural network are the ground truth of high-resolution images. We use mean square error (MSE) for the Peak Signal-to-Noise Ratio (PSNR) statistic. The objective is

$$\min \frac{1}{2N} \sum_{i=1}^N \|f(x_i) - x_i\|_2^2 \quad (17)$$

where  $f$  represents the operation of the neural network and  $x_i$  represents the input. PSNR in decibels (dB) is defined as

$$PSNR = 10 \log\left(\frac{I^2}{MSE}\right), \quad (18)$$

where  $I$  is the maximum pixel value of the data type [54]. Here, we use an 8-bit image with  $I = 255$ . The training data are generated from  $300 \times 300$  pixel images. We use the Python package *skimage* to down-scale the images at measurement rates (MR) 0.25, 0.1 and 0.06 respectively. For example, for  $\text{MR} = 0.25$  which corresponds to a reduction factor of 2 on both the horizontal and vertical axes, the generated down-scaled image is  $150 \times 150$  pixels. The compressed images for  $\text{MR} = 0.1$  and  $\text{MR} = 0.06$  are  $100 \times 100$  pixels and  $75 \times 75$  pixels, respectively. We then extract overlapped patches for the network. Patches of size  $10 \times 10$  pixels are extracted from compressed images and patches of size  $20 \times 20$  pixels ( $\text{MR} = 0.25$ ),  $30 \times 30$  pixels ( $\text{MR} = 0.1$ ) and  $40 \times 40$  pixels ( $\text{MR} = 0.06$ ) are extracted from the original images. The neural networks are trained with the Adam optimizer. The learning rate is set to be  $10^{-3}$ . The minibatch size is set to be 128. We train the neural networks for 100 epochs.

Two algorithms from the literature, ReconNet [48] and O-NL-SDA [55], are implemented for comparison with our model and serve as baselines. We implement these algorithms on the same platform as our algorithms to control the performing environment. The model architectures strictly follow those of the respective authors with activation functions and parameters chosen as suggested in their works. The performances are studied on three aspects: the stability in the training progress, the reconstruction quality on testing images and the running time.

As demonstrated with the architecture in Figure 22, representations provided by the frequency domain result in more efficient outcomes with shallower architecture. Herein, we study the progress of convergence for different algorithms to confirm the above statement. Figure 23 shows the decay of MSE over training epochs for ReconNet, ShallowNet, SparseSnet and SparseFnet at different measurement rates. For all measurement rates, the convergence progress of ReconNet shows the largest fluctuation. As the measurement rate decreases from 0.25 to 0.06, the instability of ReconNet becomes more severe. On the contrary, our models, which are in the frequency domain, have smoother MSE decay curves. Although SparseSnet experiences slow convergence for a high measurement rate of 0.25, SparseFnet has exceptional performance with rapid and stable convergence at all measurement rates.

Several experiments are conducted using standard images. Six testing images are reconstructed at  $\text{MR} = 0.25, 0.1$  and  $0.06$ . In addition to baseline models and our proposed models, we adapt the deep architecture of ReconNet’s to our frequency models SparseSnet and SparseFnet which we name D-SparseSnet and D-SparseFnet. If the performance of these deeper versions does not outperform their original shallower versions, then we can conclude that the frequency domain and the shallower architecture result in better performance when used together.

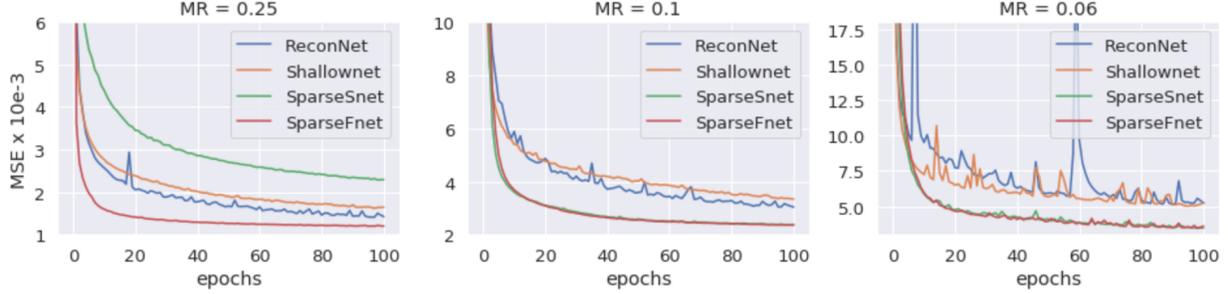


Figure 23: Decay of training MSE( $\times 10^{-3}$ ) for different algorithms at different measurement rates. The baseline model ReconNet shows the largest fluctuation as measurement rate decreases, while SparseFnet has rapid and stable convergence progress in the frequency domain.

Figure 24 shows the outcomes from different algorithms at  $MR = 0.1$ . The differences in reconstruction quality can be recognized visually. SparseSnet and SparseFnet give sharper details while models in the space domain suffer from blurring effects. Table 3 presents the testing PSNR (in dB) from the seven algorithms at all three measurement rates. For all six images, the highest PSNR are achieved by models in the frequency domain. Deeper architecture in the frequency domain performs the best in two out of the eighteen experimental trials, with the shallower architecture performing best on the remaining sixteen trials. The performances of SparseSnet and SparseFnet are similar with slight differences in general. See Table 3 for details.

Table 2 shows the running time comparison for reconstructing a  $300 \times 300$  pixel image. We use a 2.6 GHz Intel core i7-8850H CPU with 16 GB memory to run the implementations. At each measurement rate, the processing time reported for each algorithm is an average of 6 testing trials. Comparing between ReconNet and ShallowNet, D-SparseSnet and SparseSnet and D-SparseFnet and SparseFnet, the shallower architecture reduces the running time significantly. On the other hand, the superior performance of SparseSnet and SparseFnet was demonstrated in the previous section along with the slight differences between their testing PSNR. Here, we show that SparseFnet is the optimal choice, considering both accuracy and running time. Taking representations in the frequency domain as input, the major difference between SparseSnet and SparseFnet is that errors in the space domain are used to train SparseSnet, while they are not used to train SparseFnet. The operation for SparseFnet is more straight forward while SparseSnet needs to fulfill the projection within the network. Consequently, SparseFnet runs faster than SparseSnet. This point is confirmed in testing results shown in Table 2. At all measurement rates, the processing time of SparseSnet is greater than the processing time of SparseFnet.

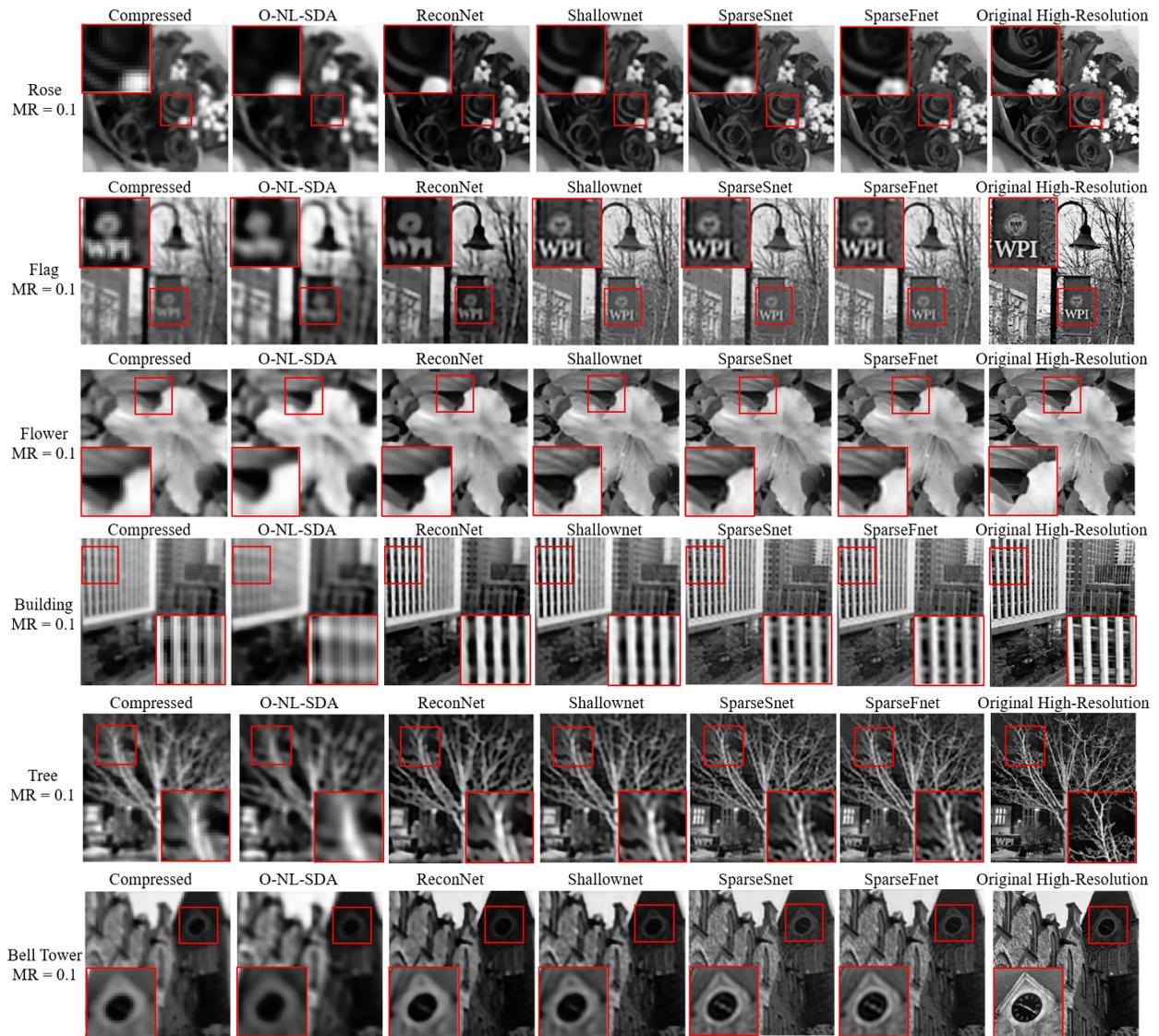


Figure 24: Reconstructed testing images with different algorithms at measurement rate of 0.1. Sharper details are revealed by our proposed SparseSnet and SparseFnet.

### 3.3 ESnet: Ensemble Super-resolution for CNNs in Transform Domains

In the study of SparseFnet, we demonstrated that the sparsifying representative of image signal could provide more efficient information to the neural network and help to overcome the challenge of small training data for image super-resolution problems. In addition to the Fourier transformation, there are other choices of transform domains. In this section, we will introduce various types of sparsifying transformations to further enrich the information which could be provided to the neural network. This way, more robust algorithms are developed for a general solution based on different image content and applications.

The Fourier transformation and wavelet transformations are competitive choices for image processing problems. A large selection of wavelet families and bandwidths for wavelet transformations exists, which allows for the optimal transform selection based on specific image content. Since our research is application-oriented, we would like to build a model which not only improves the reconstruction quality of regular images in different categories (i.e., portraits, scenes, and cartoons) but also can adapt to the areal density maps of MIRRORALON sheet material as well. Therefore, we start with the basic structure of SparseFnet, and we introduce the ability to use multiple wavelet transform domains into the framework.

In real-world applications, there are many cases where only a small collection of data is available. It is difficult to understand the distribution of images *a priori* as the sample may not be representative of the entire population. Moreover, the compressive measurements are complex and diverse according to the application field. To overcome the limitation of small training data sets and the complexity of the compressive measurements, we propose an ensemble CNN for image super-resolution problems. Including the frequency domain (DCT) and multiple wavelet transform domains, allows this model to not only provide a

Table 2: Average running time (in seconds) for reconstructing a  $300 \times 300$  image at different measurement rates. Shallow architecture reduces running time approximately by half. SparseFnet has lower running time between the best performing models (SparseSnet and SparseFnet).

|              | MR=0.25 | MR=0.1 | MR=0.06 |
|--------------|---------|--------|---------|
| ReconNet     | 19.08   | 19.46  | 18.47   |
| Shallownet   | 9.80    | 9.06   | 8.34    |
| D-SparseSnet | 28.68   | 24.19  | 23.31   |
| D-SparseFnet | 19.69   | 17.62  | 17.35   |
| SparseSnet   | 18.04   | 14.64  | 14.31   |
| SparseFnet   | 9.23    | 9.22   | 8.62    |

Table 3: Testing PSNR (in dB) with different algorithms at different measurement rates. Models in the frequency domain (D-SparseSnet, D-SparseFnet, SparseSnet and SparseFnet) yield better reconstruction quality in all the cases comparing with models in the space domain (O-NL-SDA, ReconNet and Shallownet). Shallow architecture (SparseSnet and SparseFnet) outperforms deep architecture (D-SparseSnet, D-SparseFnet) in 16 out of 18 trails.

| Testing Images | Algorithms   | MR=0.25      | MR=0.1       | MR=0.06      |
|----------------|--------------|--------------|--------------|--------------|
| Rose           | O-NL-SDA     | 25.22        | 19.06        | 14.25        |
|                | ReconNet     | 30.76        | 25.63        | 22.30        |
|                | Shallownet   | 30.83        | 26.44        | 22.92        |
|                | D-SparseSnet | 33.64        | 29.19        | 25.36        |
|                | D-SparseFnet | 33.20        | 28.80        | 25.86        |
|                | SparseSnet   | <b>33.73</b> | <b>29.25</b> | 26.08        |
|                | SparseFnet   | 33.46        | 29.13        | <b>26.27</b> |
| Flag           | O-NL-SDA     | 9.27         | 5.62         | 3.05         |
|                | ReconNet     | 14.02        | 8.78         | 5.84         |
|                | Shallownet   | 13.75        | 10.32        | 7.05         |
|                | D-SparseSnet | 17.62        | 11.92        | 8.69         |
|                | D-SparseFnet | 17.72        | 11.96        | 9.19         |
|                | SparseSnet   | 17.85        | 12.11        | 9.32         |
|                | SparseFnet   | <b>18.19</b> | <b>12.12</b> | <b>9.33</b>  |
| Flower         | O-NL-SDA     | 26.57        | 20.20        | 17.51        |
|                | ReconNet     | 30.47        | 25.40        | 19.47        |
|                | Shallownet   | 31.00        | 31.96        | 22.76        |
|                | D-SparseSnet | <b>32.67</b> | 29.16        | 25.77        |
|                | D-SparseFnet | 31.74        | 28.69        | <b>28.91</b> |
|                | SparseSnet   | 30.43        | 29.36        | 28.06        |
|                | SparseFnet   | 31.67        | <b>29.39</b> | 28.06        |
| Building       | O-NL-SDA     | 13.85        | 7.36         | 5.61         |
|                | ReconNet     | 20.55        | 14.94        | 8.70         |
|                | Shallownet   | 20.51        | 16.01        | 9.20         |
|                | D-SparseSnet | 23.47        | 18.74        | 13.31        |
|                | D-SparseFnet | 23.78        | 19.03        | 13.50        |
|                | SparseSnet   | 24.57        | <b>19.28</b> | 14.02        |
|                | SparseFnet   | <b>24.72</b> | 19.25        | <b>14.04</b> |
| Tree           | O-NL-SDA     | 15.05        | 12.66        | 10.78        |
|                | ReconNet     | 19.70        | 15.78        | 13.86        |
|                | Shallownet   | 19.85        | 15.61        | 13.90        |
|                | D-SparseSnet | 23.23        | 18.45        | 15.48        |
|                | D-SparseFnet | 23.18        | 18.33        | 15.39        |
|                | SparseSnet   | 23.73        | <b>18.57</b> | 15.76        |
|                | SparseFnet   | <b>23.78</b> | 18.53        | <b>15.84</b> |
| Bell Tower     | O-NL-SDA     | 19.66        | 16.33        | 12.94        |
|                | ReconNet     | 24.26        | 19.88        | 17.52        |
|                | Shallownet   | 24.03        | 20.60        | 17.86        |
|                | D-SparseSnet | 26.73        | 22.46        | 19.39        |
|                | D-SparseFnet | 26.62        | 22.36        | 19.73        |
|                | SparseSnet   | 26.62        | <b>22.59</b> | <b>19.97</b> |
|                | SparseFnet   | <b>26.82</b> | 22.53        | 19.93        |

universal solution for choosing between DCT and the different wavelet transformations but also gives a more robust result when compared with other advanced models in the original space domain. The neural network ensemble can learn the properties of the specific type of sub-image patches and matches the patches to a suited transform domain.

We demonstrated with SparseFnet that a sparse domain for CNN-based super-resolution can be effective. We assume that the sparse signal in a particular transform domain is more representative when compared with the original space domain when the collection of available data is small. However, the selection of transformation is crucial and varies for different compressive measurements. The transform domain determines not only the computational complexity but also the reconstruction quality [21]. As image super-resolution neural networks classically work on overlapping sub-image patches instead of the entire image, the image versus transform domain mismatch problems may be increased with the wrong selection of transform. Additionally, the frequency distribution of a sub-image patch may be far from the frequency distribution of another sub-image patch in the same image. Therefore, making a single choice among the large family of transformations based on image content and application cannot necessarily meet the demand of real-world images.

We propose an ensemble CNN for single image super-resolution problems, and we call it ESnet. The ESnet considers reconstructions from multiple transform domains and assembles the results using different weights. The model learns the characteristics of sub-image patches and assigns different weights based on the contents of patches. The overall result should be robust and have better quality than any of the reconstructions from a single transform domain.

### **The architecture of ESnet**

The architecture of ESnet is shown in Figure 25. The model first contains a part of basic image processors in different transform domains, and then another neural network that works as a combiner. Before training the combiner, we train the individual processors in the DCT and wavelet domains from three different families. With the resulting DCT and wavelet representations, the basic image processors increase the resolution of overlapped small image patches which are extracted from the entire image. The combiner then takes the ground truth of the high-resolution patch as input and learns the performances of different domains with respect to the reconstruction quality. A weight is assigned to each individual processor by the combiner. High-resolution image patches from different domains are simplified for final output.

The settings of the basic image processors follow what is illustrated by Figure 26. The shallow architecture is inspired by SparseFnet, where we modify two versions for the DCT

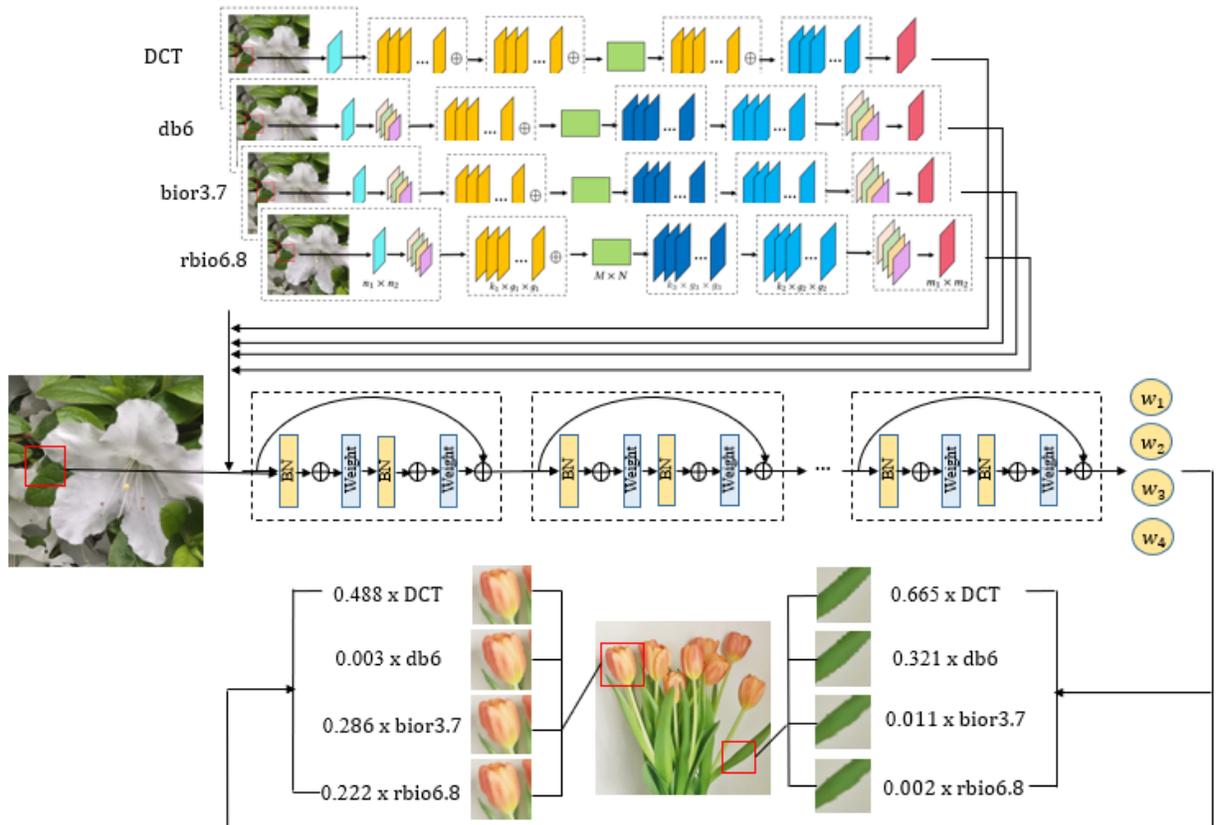


Figure 25: The architecture for the ensemble model ESnet. Four individual processors in different transform domains are utilized as image basic processors. A combiner using the framework of ResNet is trained to assign weights to individual processors based on the characteristic of image patches.

domain and the wavelet domain respectively. In the DCT domain, a dense layer is in the middle of four convolutional layers to map from low-dimensional feature space to high-dimensional feature space. The first three convolutional layers have sixty-four  $3 \times 3$  kernels followed by the ReLU activation function, and the last convolutional layer has only one  $5 \times 5$  kernel. In the wavelet transform domains, the multi-channel wavelet representation first passes through a convolutional layer with sixty-four  $3 \times 3$  kernels. A ReLU activation function is placed before the dense layer to introduce a non-linearity measurement. After the linear mapping, the features are processed by two hundred fifty-six  $8 \times 8$  kernels and then one  $5 \times 5$  kernel in the last convolutional layer. The individual basic processors with DCT and wavelet transformations are trained purely in the transform domains and are never exposed to the space domain. A deep residual neural network called ResNet [56] is adapted to be the combiner. ResNet was originally designed for image recognition problems. Instead of outputting classification labels, we modify the model to output weights. The target weights for input image patches are defined as

$$\begin{aligned} \mathbf{w} &= \{w_1, w_2, w_3, w_4, w_5\} \\ w_i &= \frac{MSE_{max}}{MSE_i} - 1, \quad i = 1, \dots, 5, \quad \sum_{i=1}^5 w_i = 1 \end{aligned} \quad (19)$$

where  $MSE_i$ s are mean square errors from individual processors and  $MSE_{max}$  is the largest value among them. The objective for the combiner is

$$\begin{aligned} \epsilon &= \Psi_\eta(x) \\ \min_{\eta} \|\mathbf{w} - \epsilon\|_2^2 \end{aligned} \quad (20)$$

where  $\Psi_\eta(\cdot)$  denotes the operation of the combiner,  $x$  represents the input image patch and  $\epsilon = \{\epsilon_1, \epsilon_2, \epsilon_3, \epsilon_4, \epsilon_5\}$  is the estimate of the weights.

The training pseudo-code for ESnet is provided in Algorithm 2.

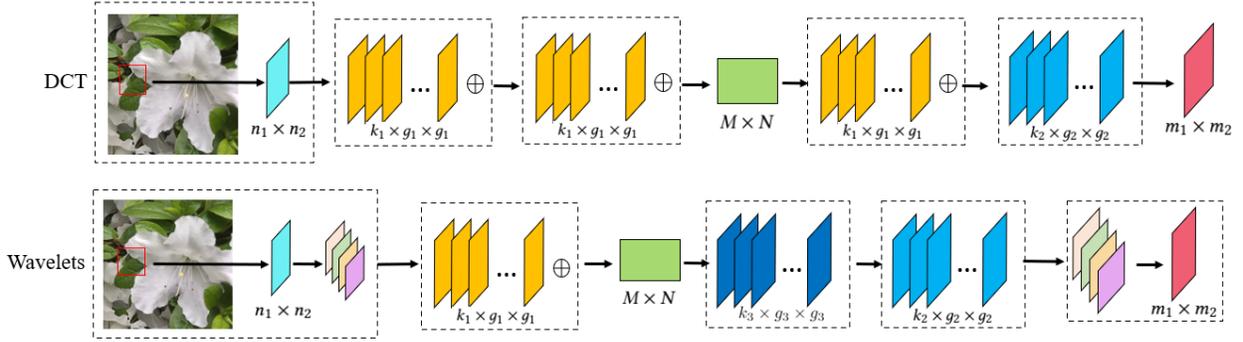


Figure 26: The architecture for basic image processors in DCT domain and wavelet transform domains. In DCT domain, a dense layer is in the middle of four convolutional layers to linearly map the features from low-dimensional space into high-dimensional space. Relu is applied after the first three convolutional layers. In wavelet transform domains, the algorithm is performed on multi-channel wavelet representations. A dense layer is applied after the first convolutional layers to linearly map the features from low-dimensional space into high-dimensional space. Relu is applied only to the first convolutional layer.

---

**Algorithm 2** ESnet

---

**Input:**  $x \in R^{n_1 \times n_2}$

Initialize  $\Phi_i$  in individual SparseFnets with randomly initialized parameters,  $i = 1, \dots, 5$ .

**For**  $i = 1, \dots, 5$

$$h_i = \Phi_i(x)$$

Minimize  $\|\Omega_i - \Phi_i(x)\|_2^2$  using back-propagation.

**end for**

**Return**  $\mathbf{h} = \{h_1, h_2, h_3, h_4, h_5\}, MSE_i, i = 1, \dots, 5$ .

Set  $\mathbf{w}$  with equation (5) for the combiner.

Initialize  $\Psi_\eta$  with randomly initialized parameters.

**While** epoch = True

**for** each iteration

$$\epsilon = \Psi_\eta(x)$$

  Minimize  $\|\mathbf{w} - \epsilon\|_2^2$  using back-propagation.

**end for**

**end while**

**Return**  $\epsilon$

**Output:**  $y = \epsilon \cdot \mathbf{h}, y \in R^{m_1 \times m_2}$

---

## ESnet on ordinary image reconstruction

Our target is to obtain an algorithm that picks the most suited transform domains with small numbers of the image signal for complex compressive measurements. We use standard benchmark image data sets as training and testing data here. One out of 800 high-resolution images from the DIV2K data set [57] is randomly drawn as the training data for each experiment. Testing results are compared on data sets Set5 and Set14 which consist of natural scenes. In previous studies, cubic interpolation is often chosen to be the standard compressive measurement. However, the up-sampling convolutional layer within previous models is mathematically similar to cubic interpolation. We believe it is more faithful to consider more measurements. To present the complexity and diversity of compressive measurements in real-world applications, twelve down-scaling procedures are carried out with different interpolation methods, Gaussian blur models, and Gaussian pyramid degradation as shown in Table 11. We use packages OpenCV and scikit-image in Python to implement the measurements. Four down-scaling procedures are conducted with super-resolution factors of 2, 4, and 6 respectively. In other words, a  $100\times 100$  image would be reconstructed to  $200\times 200$ ,  $400\times 400$ , and  $600\times 600$ . Each of the twelve down-scaling procedures consists of multiple measurements. Suppose we have a  $400\times 400$  high-resolution image. In down-scaling procedure No. 5 with a factor of 4, the image is compressed into  $200\times 200$  with linear interpolation first. Then, a Gaussian blur is added to smooth the image. Finally, the image is compressed into  $100\times 100$  with the nearest interpolation.

Six algorithms are implemented as baselines. SRResNet [12] and EDSR [13] are high-performance benchmark deep residual neural networks in the original space domain from the literature. Basic image processors illustrated in Figure 26 are individually performed in four different transform domains. We include DCT for Fourier domain, Daubechies [41], biorthogonal, and reverse biorthogonal for wavelet domains [23, 42, 43]. Due to the small size of the training data set, we use the same method to process image patches in the experiments. SRResNet and EDSR are adapted to corresponding input and output image shapes.

Twelve experiments on two testing data sets at three different super-resolution factors are conducted to compare the performance of ESnet with baseline models. To demonstrate the effectiveness of ensemble over selecting a most-suited sparse domain, we choose DCT, db6 from the Daubechies wavelet family, bior5.5 from the biorthogonal wavelet family, and rbio6.8 from reverse biorthogonal wavelets family for individual baseline models as well as for basic processors in ESnet. Packages SciPy and PyWavelets in Python are used to implement the transformations. Numerical PSNR (in dB) values from baseline models, individual processors, and ESnet are listed in Table 16. The best and the second best PSNR values are labeled in red and blue respectively.

Table 4: 12 down-scaling procedures with multiple measurements. Lanczos: Lanczos interpolation; Cubic: cubic interpolation; Nearest: nearest interpolation; Linear: linear interpolation; Gaussian: Gaussian Pyramid; GB: Gaussian Blur.

| <b>Super-resolution Factors</b> | <b>Down-scaling Procedure No.: Measurements</b>   |
|---------------------------------|---|
| x2                              | 1: Lanczos x1.4 + GB + Nearest x1.4<br>2: Gaussian x2+ GB<br>3: Gaussian x1.4 + Nearest x1.4 + GB<br>4: Gaussian x1.4 + GB + Gaussian x1.4 + GB |
| x4                              | 5: Linear x2 + GB + Nearest x2<br>6: Lanczos x2 + Nearest x2 + GB<br>7: Gaussian x4 + GB<br>8: Gaussian x2 + GB + Gaussian x2 + GB              |
| x6                              | 9: Lanczos x2 + GB + Linear x3 + GB<br>10: Gaussian x6 + GB<br>11: Gaussian x3 + GB + Cubic x2 + GB<br>12: Gaussian x2 + GB + Gaussian x3 + GB  |

The best-suited transform domain varies from different testing data sets and different compressive measurements. It shows that selecting a single type of sparsifying transformation based upon the image content and the specific application cannot guarantee an optimal solution. In contrast, the reconstruction quality from our ensemble model is always better than that of individual processors. ESnet combines reconstructed sub-image patches based on the performances of different transform domains. The contribution ratio (weight) of each domain is customized for each patch and specific compressive measurements. By making different selections for different sub-image patches, the overall outcome is more accurate than that from the best-suited sparsifying transformation for the entire image.

The performance of SRResNet and EDSR varies from different procedures. They depend on the randomly drawn small training image as well as the 'unknown' compressive measurements. For instance, SRResNet has competing performance for procedures No. 1, 4, 6, 9, 10, and 11. However, procedures No. 2, 3, 7, and 12, give low PSNR values, even lower than individual basic processors in single transform domains. The instability of small training data sets and complex compressive measurements is also shown between SRResNet and EDSR. SRResNet generates the second-best results while EDSR has the worst results for procedure No. 6, but for procedure No. 7, the situation is reversed. In contrast, the ensemble model ESnet has a more stable performance. Figure 30 plots the performance

Table 5: Testing PSNR (in dB) with different models for different Super-resolution Factors (SF). DCT, db6, rbio and bior are individual basic image processors in transform domains. The measurements for each procedure can be found in Table 11 with the corresponding Procedure Number (PN). Every measurement is tested on Set 5 and Set 14. For each of the 24 procedures, the red reflects the highest PSNR values and the blue reflects the second highest PSNR values. The ensemble model ESnet outperforms comparing models in 18 procedures.

| SF | PN | Test Set | SRResNet      | EDSR          | DCT           | db6    | rbio          | bior          | ESnet         |
|----|----|----------|---------------|---------------|---------------|--------|---------------|---------------|---------------|
| x2 | 1  | Set5     | <b>24.925</b> | 24.512        | 24.518        | 24.322 | 24.328        | 24.206        | <b>24.562</b> |
|    |    | Set14    | <b>22.973</b> | <b>22.625</b> | 22.526        | 22.471 | 22.439        | 22.378        | 22.595        |
|    | 2  | Set5     | 25.503        | <b>26.480</b> | 26.219        | 26.286 | 25.803        | 25.812        | <b>26.564</b> |
|    |    | Set14    | 23.540        | <b>24.131</b> | 23.838        | 24.011 | 23.434        | 23.691        | <b>24.155</b> |
|    | 3  | Set5     | 23.907        | 23.367        | 23.675        | 24.130 | <b>24.260</b> | 23.638        | <b>24.262</b> |
|    |    | Set14    | 22.265        | 21.970        | 22.120        | 22.207 | <b>22.468</b> | 22.161        | <b>22.717</b> |
|    | 4  | Set5     | 23.697        | 23.632        | 23.013        | 23.100 | <b>23.748</b> | 23.338        | <b>23.808</b> |
|    |    | Set14    | <b>22.105</b> | 22.040        | 21.571        | 21.774 | 22.062        | 21.871        | <b>22.186</b> |
| x4 | 5  | Set5     | 22.999        | 23.032        | <b>23.103</b> | 22.479 | 22.606        | 22.547        | <b>23.211</b> |
|    |    | Set14    | 21.209        | 21.325        | <b>21.352</b> | 20.862 | 20.997        | 20.967        | <b>21.378</b> |
|    | 6  | Set5     | <b>23.171</b> | 22.543        | 23.117        | 22.711 | 22.943        | 22.708        | <b>23.226</b> |
|    |    | Set14    | <b>21.142</b> | 20.885        | 21.133        | 20.909 | 20.827        | 20.935        | <b>21.152</b> |
|    | 7  | Set5     | 24.248        | <b>26.466</b> | 23.732        | 24.521 | 24.569        | 24.841        | <b>24.683</b> |
|    |    | Set14    | 22.255        | <b>23.857</b> | 22.008        | 22.548 | 22.621        | 22.794        | <b>22.728</b> |
|    | 8  | Set5     | 23.622        | 23.444        | 22.860        | 24.038 | 24.064        | <b>24.209</b> | <b>24.355</b> |
|    |    | Set14    | 22.023        | 21.843        | 21.356        | 22.218 | 22.236        | <b>22.420</b> | <b>22.445</b> |
| x6 | 9  | Set5     | <b>21.407</b> | 21.178        | 20.951        | 20.368 | 20.706        | 20.751        | <b>21.431</b> |
|    |    | Set14    | 20.416        | <b>20.447</b> | 20.109        | 19.722 | 19.928        | 19.882        | <b>20.448</b> |
|    | 10 | Set5     | <b>23.282</b> | <b>23.407</b> | 22.036        | 23.036 | 23.022        | 22.980        | 23.213        |
|    |    | Set14    | 21.636        | <b>21.820</b> | 20.832        | 21.512 | 21.517        | 21.548        | <b>21.707</b> |
|    | 11 | Set5     | <b>22.278</b> | 22.264        | 19.835        | 21.570 | 21.788        | 21.585        | <b>22.346</b> |
|    |    | Set14    | <b>21.071</b> | 21.012        | 19.143        | 20.541 | 20.660        | 20.533        | <b>21.106</b> |
|    | 12 | Set5     | 19.070        | 19.638        | 17.912        | 20.051 | 20.159        | <b>20.227</b> | <b>20.623</b> |
|    |    | Set14    | 18.783        | 18.788        | 17.457        | 19.307 | 19.417        | <b>19.426</b> | <b>19.739</b> |

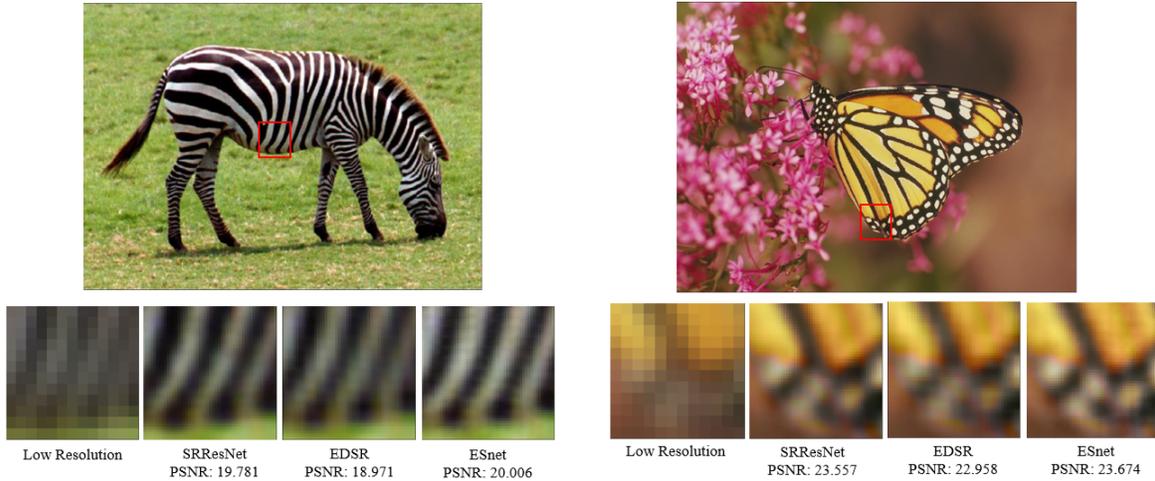


Figure 27: Reconstructed testing images from procedure No.8. The super-resolution factor is 4. ESnet provides the highest PSNR values and the sharpest details.

ranking of ESnet, SRResNet, and EDSR on Set 5 and Set 14. For both testing data sets, the benchmark comparing models show instability from procedure to procedure. Our proposed ESnet is more robust and always yields high rankings. On the other hand, visual differences in reconstruction quality among baseline models and ESnet can be recognized in Figure 27. Two images from procedure No. 8 are given. The enlarging windows present some outcome of instability. When the 'unknown' compressive measurement is very complex, and the small training data set does not well represent the entire population, benchmark models may suffer from blurring and deformation. As ESnet considers the different separation of frequencies in transform domains, it robustly recovers better details.

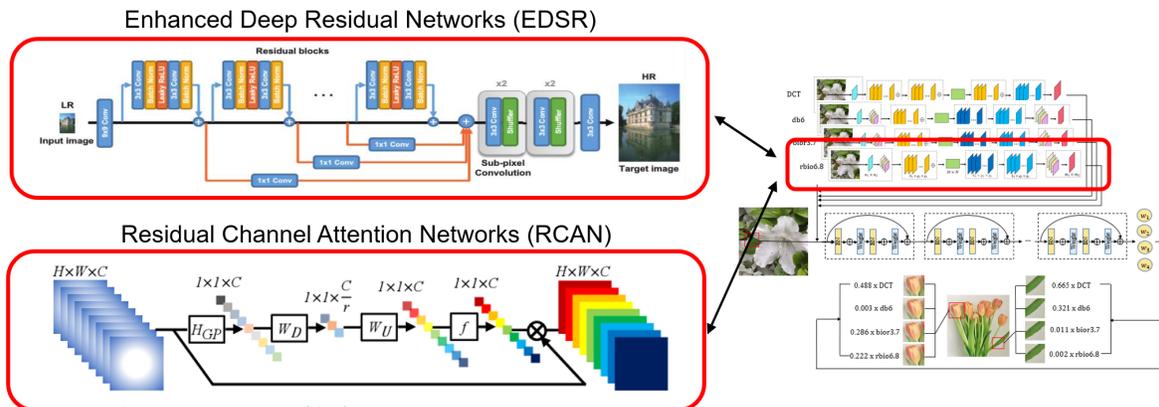


Figure 28: The ablation study of ESnet. Replace the SparseFnet with Benchmark algorithms EDSR and RCAN to measure the impact of the basic image processor on the performance of ESnet.

### The ablation study of ESnet

In ESnet, multi-channel representations are used in the basic image processor using wavelet transform domains to improve the efficiency of extracting features. Starting from this point, we look for more efficient methods for multi-channel feature extraction. The first attempt to improve the performance of ESnet is to replace the SparseFnet with benchmark algorithms as the basic image processor. As illustrated in Figure 32, experiments are conducted with EDSR and RCAN. Since the deep residual blocks in EDSR and RCAN suffer from gradient explosion when the training data is extremely small, we reduce the number of residual blocks to be 20. If EDSR can generate more accurate results than single domain SparseFnet, then replacing SparseFnet with EDSR in ESnet should help with the performance.

As EDSR has better performance than individual shallow basic image processors in single transform domains (refer to Table 16), the deep ESnet model has the potential for outperforming regular ESnet. However, the experimental results shown in Table 6 suggest that shallow architecture is actually superior in the presence of small amounts of training data.

In this table, shallow models are in original architectures, deep models are EDSRs performing on wavelet transform domains, and EDSR-embedded ESnet. The best PSNR values, compared between shallow and deep architectures, are labeled in orange in the table. For both basic image processors in a single transform domain and ensemble models in multiple domains, the shallow architecture has better performance. The performance of EDSR drops when it performs on the sparse transform domain instead of the space domain. For both single sparse domains db6 and bior5.5, EDSR has lower PSNR values than our original SparseFnet. This may not only be due to the deep architecture but also be related to the nature of sparse representatives. EDSR may pay more attention to the low-frequency

Table 6: Testing PSNR (in dB) for down-scaling Procedure No. (PN) 5-8 with shallow and deep architectures from single wavelet domains and ESnet. Shallow models are the same as illustrated in the previous section. Deep models replace the basic image processor in Figure 32 with EDSR.

|           |                 | <b>db6</b>     |             | <b>bior5.5</b> |             | <b>ESnet</b>   |             |
|-----------|-----------------|----------------|-------------|----------------|-------------|----------------|-------------|
| <b>PN</b> | <b>Test Set</b> | <b>shallow</b> | <b>deep</b> | <b>shallow</b> | <b>deep</b> | <b>shallow</b> | <b>deep</b> |
| 5         | Set5            | 22.479         | 22.44       | 22.606         | 22.536      | 23.211         | 23.092      |
|           | Set14           | 20.862         | 20.976      | 20.997         | 20.987      | 21.378         | 21.359      |
| 6         | Set5            | 22.711         | 22.515      | 22.708         | 22.658      | 23.226         | 23.062      |
|           | Set14           | 20.909         | 20.836      | 20.935         | 20.92       | 21.152         | 21.174      |
| 7         | Set5            | 24.521         | 24.257      | 24.841         | 24.701      | 24.683         | 24.641      |
|           | Set14           | 22.548         | 22.332      | 22.794         | 22.621      | 22.728         | 22.603      |
| 8         | Set5            | 24.064         | 23.574      | 24.209         | 23.664      | 24.355         | 24.281      |
|           | Set14           | 22.236         | 21.977      | 22.42          | 21.996      | 22.445         | 22.473      |

features. With the separated frequency representatives from transform domains, EDSR may fail to extract enough information from the high-frequency part.

Traditionally, channels are treated equally in the neural network. However, some channels containing abundant low-frequency information may hinder the network from focusing on other channels. Zhang et al [58] developed Residual Channel Attention Networks (RCAN) to address this problem. In general, they use deep residual blocks and skip connections to preserve both low-frequency and high-frequency information. The main network focuses on high-frequency information, while some residual blocks pass the low-frequency information through multiple skip connections. Considering the internal dependencies among different channels, RCAN rescales channel-wise features to preserve the balance of the energy content of the underlying channels. In our case, the approximation channels within the four wavelet detail sub-matrix (refer to Figure 14) are primarily collecting high-frequency information, while the other three channels are all built for low-frequency information. Although the high-frequency loss in our shallow architecture is not as severe as in deep neural networks, we adapt a shallower version of RCAN in ESnet to explore the possibility of improving the reconstruction quality. We replace the shallow processors with 56-layer RCANs. Several experiments are conducted for down-scaling procedures No.5-No.8 in Table 11. The testing PSNR values are shown in Table 7. Unfortunately, only 2 out of 8 experiments show that the RCAN embedded ESnet has better performance than ESnet. Although the attention mechanism is helpful for regular image super-resolution networks with abundant training data, it does not help ESnet to extract wavelet features more efficiently. This suggests

Table 7: Testing PSNR (in dB) for RCAN embedded ESnet and ESnet.

|    |   | Test Set | RCAN-ESnet    | ESnet  |
|----|---|----------|---------------|--------|
| x4 | 5 | Set5     | 23.092        | 23.211 |
|    |   | Set14    | 21.359        | 21.378 |
|    | 6 | Set5     | 23.061        | 23.226 |
|    |   | Set14    | <b>21.174</b> | 21.152 |
|    | 7 | Set5     | 24.642        | 24.683 |
|    |   | Set14    | 22.603        | 22.728 |
|    | 8 | Set5     | 24.280        | 24.355 |
|    |   | Set14    | <b>22.473</b> | 22.445 |

the robust performance of shallow architecture for the small data image super-resolution problem.

As demonstrated in the small-scale ablation study the shallower architecture is more sufficient for not only the single domain models but also the ensemble model. We attempt to improve several other aspects of our ESnet. There are two algorithms developed focusing on multi-channel sparse representatives and ensembles respectively. In particular, ESnet has a complex architecture, and we believe that by treating channels in a more uniform way the architecture can be simplified while maintaining good performance. The reconstruction module and the ensemble module are trained separately and stored in different files. Herein we simplify the two-step algorithm into one-step algorithms to improve efficiency and, perhaps surprisingly, actually improve performance. In particular, the deep residual architecture in ESnet is removed since we believe the deep architecture cannot be well-trained on a small sample. The new models utilize multiple sparsifying representatives from different wavelet domains in a uniform way and output a high-resolution reconstruction in the space domain. Both algorithms, which we call EWnet and EnsemNet not only simplify the training procedure but also further improve the reconstruction quality.

### 3.4 EWnet: Ensemble Super-resolution CNN in Wavelet domains

The ESnet algorithm decides the best choice of transform domain for each sub-image patch and combines the patches to provide a general optimal solution for the entire image. However, the architecture has the potential to be optimized and here we attempt to improve the design

of ESnet in two aspects:

- First, we analyze the performance of the basic image processors.
- Second, we simplify our original complex architecture by treating image channels in a more uniform fashion.

### **The architecture of EWnet**

In our experience, having the neural networks focus on the transform domain and not the original image domain leads to better performance. Accordingly, we chose to separately train the reconstruction module and the ensemble module in ESnet so that each reconstruction module only focuses on a single transform domain.

However, the above choice leads to a complex architecture for the ESnet model. Five independent models are trained and stored for one ESnet. As illustrated in Figure 32, there are four basic image processors performed in the DCT domain, db6, bior3.7, and rbio6.8 wavelet transform domains respectively. The combiner then makes the ensemble in the original image domain. While such a procedure leads to accurate image reconstructions, the computational cost and the time cost are also important considerations in many applications. In practice, we prefer a simplified architecture that requires only one training process and, therefore, we developed the EWnet architecture, as described here.

Figure 29 shows the architecture of EWnet. It takes multi-channel wavelet representations from three different wavelet transform domains as input and outputs high-resolution image patches in the original space domain. The algorithms are modified from the basic wavelet image processor in ESnet, which is shown in Figure 26. It contains a feature extraction layer, up-sampling layers, and a reconstruction layer. In the pre-processing stage, three 4-channel representatives from different wavelet domains are combined into a 12-channel representative. Then, a convolutional layer extracts 32 low-dimensional features from the channels by  $3\times 3$  kernels and a ReLU activation function. Next, the extracted features are mapped into high-dimensional space by 2 convolutional up-sampling layers and a linear up-sampling layer. The convolutional up-sampling layers further extract 128 features and project them into higher dimensional space for 32 features. After that, the linear up-sampling layer adjusts the dimension and maps the features into the target dimension. In the end, a convolutional layer takes 4 features from the linear layer and reconstructs the high-resolution image patch in the original space domain from 32 features. In this architecture, all the convolutional layers use  $3\times 3$  kernels.

Previously, we indicated that a shallower architecture is more robust to numerous deep layers for the small data image super-resolution problems that we focus on. EWnet does

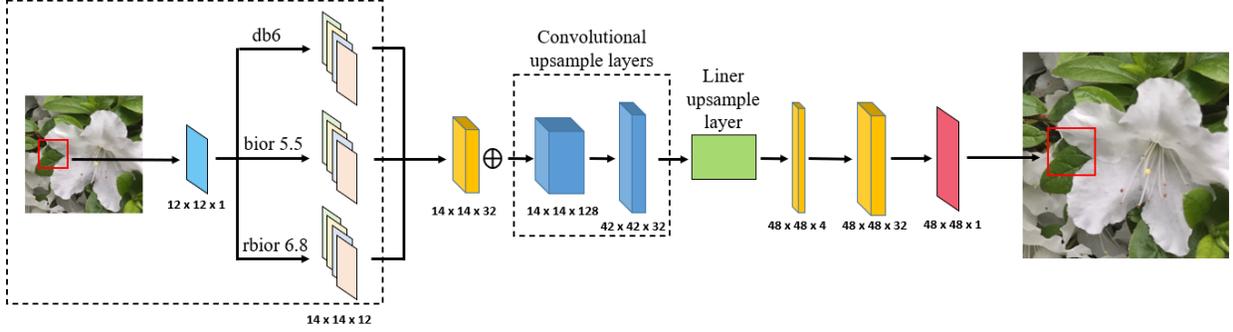


Figure 29: The architecture for the EWnet in wavelet transform domains. The algorithm is performed on multi-channel wavelet representations. It contains a convolutional feature extraction layer, two convolutional up-sampling, one linear up-sampling layer, and a reconstruction convolutional layer.  $3 \times 3$  kernels are applied.

---

**Algorithm 3** EWnet

---

**Input:**  $y_s \in R^{n_1 \times n_2}$

Pre-training process:  $x_{db} = \psi_{db}(y_s)$ ,  $x_{bior} = \psi_{bior}(y_s)$ ,  $x_{rbior} = \psi_{rbior}(y_s)$ .

Concatenate  $x_{db}$ ,  $x_{bior}$  and  $x_{rbior}$  into a multi-channel representative  $x$ .

Initialize  $\Phi_\theta$  with randomly initialized parameters.

**While** epoch = True

**for** each iteration

$y = \Phi_\theta(x)$

        Minimize  $\|\Omega - \Phi_\theta(x)\|_2^2$  using back-propagation.

**end for**

**end while**

**Return**  $y$

**Output:**  $y \in R^{m_1 \times m_2}$

---

not require a combiner, which is itself a deep residual neural network, but instead leverages a shallow convolutional structure. Since the architecture is significantly simplified, we also observe that EWnet also makes improvements to the stability of the training process and the robustness of different input images, as compared to ESnet. The training pseudo-code for EWnet is provided in Algorithm 3.

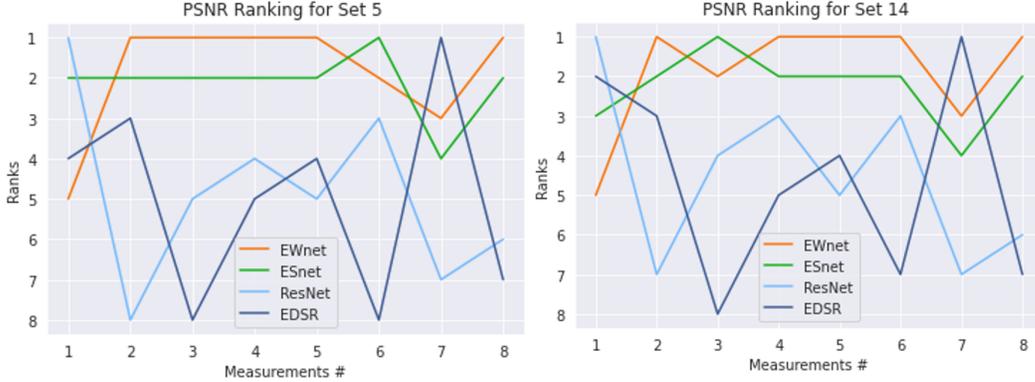


Figure 30: PSNR rankings of SRResNet, EDSR, ESnet, and EWnet referring test results in Table 16. EWnet ranks the highest overall and performs more stably than comparing methods for small training data sets.

### EWnet on ordinary image reconstruction

In the ablation study of ESnet, it is demonstrated that shallower architectures are sufficient, and even in many cases superior, for our proposed ensemble model, we next make comparisons among benchmark deep residual neural networks. In particular, we compare ESnet and EWnet with other baseline models to illustrate the efficiency. Then, we include more data sets to analyze the flexibility of EWnet on small training data.

Table 17 shows the comparisons between ESnet and EWnet from the sixteen experiments with super-resolution factors of 2 and 4. Although EWnet uses only the wavelet transform domain and misses the information from the frequency domain, its shallow architecture helps to outperform ESnet in 12 out of the 16 experiments. In addition, EWnet is more stable than ESnet. Figure 30 plots the performance ranking of EWnet, ESnet, SRResNet, and EDSR on Set 5 and Set 14. For both testing data sets, the benchmark comparing models shows instability from procedure to procedure. Our proposed ensemble models are more robust and always yield high rankings. In particular, EWnet has better performance than ESnet on average. On the other hand, visual differences in reconstruction quality among baseline models and EWnet can be recognized in Figure 31. Images from down-scaling procedure No. 8 are given. The enlarging windows present some outcome of instability. When the 'unknown' compressive measurement is very complex, and the small training data set is not well representing the entire population, benchmark models may suffer from blurring and deformation. As ESnet considers the different separation of frequencies in transform domains, it robustly recovers better details.

Table 8: Testing PSNR (in dB) with different models for different Super-resolution Factors (SF). DCT, db6, rbio, and bior are individual basic image processors in transform domains. The measurements for each procedure can be found in Table 11 with the corresponding Procedure Number (PN). Every measurement is tested on Set 5 and Set 14. For each of the 16 procedures, the red reflects the highest PSNR values and the blue reflects the second highest PSNR values. The ensemble models ESnet and EWnet yield the highest PSNR values in 12 procedures. In particular, EWnet outperforms ESnet in most cases.

| SF | PN | Test Set | SRResNet      | EDSR          | DCT    | db6    | rbio   | bior   | ESnet         | EWnet         |
|----|----|----------|---------------|---------------|--------|--------|--------|--------|---------------|---------------|
| x2 | 1  | Set5     | <b>24.925</b> | 24.512        | 24.518 | 24.322 | 24.328 | 24.206 | <b>24.562</b> | 24.419        |
|    |    | Set14    | <b>22.973</b> | <b>22.625</b> | 22.526 | 22.471 | 22.439 | 22.378 | 22.595        | 22.489        |
|    | 2  | Set5     | 25.503        | 26.480        | 26.219 | 26.286 | 25.803 | 25.812 | <b>26.564</b> | <b>26.764</b> |
|    |    | Set14    | 23.540        | 24.131        | 23.838 | 24.011 | 23.434 | 23.691 | <b>24.155</b> | <b>24.237</b> |
|    | 3  | Set5     | 23.907        | 23.367        | 23.675 | 24.130 | 24.260 | 23.638 | <b>24.262</b> | <b>24.282</b> |
|    |    | Set14    | 22.265        | 21.970        | 22.120 | 22.207 | 22.468 | 22.161 | <b>22.717</b> | <b>22.500</b> |
|    | 4  | Set5     | 23.697        | 23.632        | 23.013 | 23.100 | 23.748 | 23.338 | <b>23.808</b> | <b>23.899</b> |
|    |    | Set14    | 22.105        | 22.040        | 21.571 | 21.774 | 22.062 | 21.871 | <b>22.186</b> | <b>22.250</b> |
| x4 | 5  | Set5     | 22.999        | 23.032        | 23.103 | 22.479 | 22.606 | 22.547 | <b>23.211</b> | <b>23.261</b> |
|    |    | Set14    | 21.209        | 21.325        | 21.352 | 20.862 | 20.997 | 20.967 | <b>21.378</b> | <b>21.437</b> |
|    | 6  | Set5     | 23.171        | 22.543        | 23.117 | 22.711 | 22.943 | 22.708 | <b>23.226</b> | <b>23.177</b> |
|    |    | Set14    | 21.142        | 20.885        | 21.133 | 20.909 | 20.827 | 20.935 | <b>21.152</b> | <b>21.194</b> |
|    | 7  | Set5     | 24.248        | <b>26.466</b> | 23.732 | 24.521 | 24.569 | 24.841 | 24.683        | <b>24.705</b> |
|    |    | Set14    | 22.255        | <b>23.857</b> | 22.008 | 22.548 | 22.621 | 22.794 | 22.728        | <b>22.775</b> |
|    | 8  | Set5     | 23.622        | 23.444        | 22.860 | 24.038 | 24.064 | 24.209 | <b>24.355</b> | <b>24.616</b> |
|    |    | Set14    | 22.023        | 21.843        | 21.356 | 22.218 | 22.236 | 22.420 | <b>22.445</b> | <b>22.621</b> |

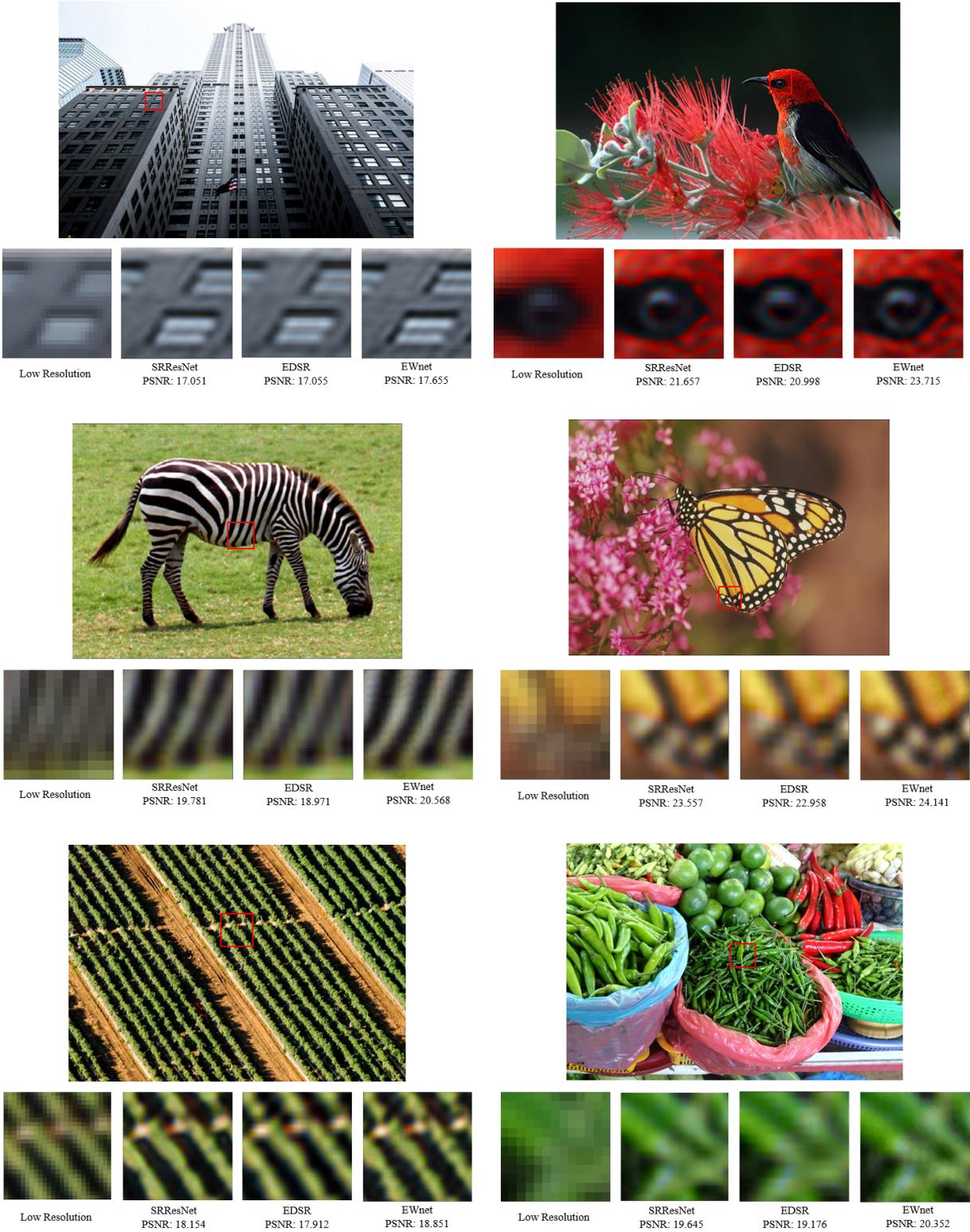


Figure 31: Reconstructed testing images from procedure No.8. The super-resolution factor is 4. ESnet provides the highest PSNR values and the sharpest details.

Table 9: Testing PSNR (in dB) with different models for a small dataset with 50 patches.

| Super-resolution Factors | Down-scaling Procedure No. | Test Set | SRResNet      | EDSR          | EWnet         |
|--------------------------|----------------------------|----------|---------------|---------------|---------------|
| x2                       | 1                          | Set5     | 24.477        | <b>25.042</b> | 24.550        |
|                          |                            | Set14    | 22.566        | <b>22.955</b> | 22.605        |
|                          | 2                          | Set5     | 29.470        | 29.186        | <b>29.805</b> |
|                          |                            | Set14    | 26.278        | 25.970        | <b>26.524</b> |
|                          | 3                          | Set5     | 24.402        | 24.311        | <b>24.730</b> |
|                          |                            | Set14    | 22.542        | 22.510        | <b>22.832</b> |
|                          | 4                          | Set5     | 23.291        | 23.114        | <b>23.726</b> |
|                          |                            | Set14    | 21.709        | 21.623        | <b>22.124</b> |
| x4                       | 5                          | Set5     | 23.507        | <b>23.796</b> | 23.474        |
|                          |                            | Set14    | 21.611        | <b>21.676</b> | 21.596        |
|                          | 6                          | Set5     | <b>24.020</b> | 23.224        | 23.179        |
|                          |                            | Set14    | <b>21.917</b> | 21.402        | 21.263        |
|                          | 7                          | Set5     | 24.851        | 24.607        | <b>24.893</b> |
|                          |                            | Set14    | 22.779        | 22.619        | <b>22.789</b> |
|                          | 8                          | Set5     | 24.544        | 23.780        | <b>24.590</b> |
|                          |                            | Set14    | 22.571        | 22.050        | <b>22.574</b> |

To further test the superiority of EWnet for specifically small data, two extra small training data sets are considered. The half-image data set contains 50 image patches, and the two-images data set contains 200 image patches. Comparisons between EWnet, SRResNet and EDSR are given in Table 9 and Table 10. The 50-patches set is extremely small that covers a  $120 \times 60$  pixels area. It requires high flexibility and stability. Among all the eight down-scaling procedures, EWnet has better performance than the benchmark algorithms for five procedures. Similarly, EWnet yields higher PSNR values in eleven out of sixteen experiments when we enlarge the training data set to 200 patches.

Table 10: Testing PSNR (in dB) with different models for a small dataset with 200 patches.

| Super-resolution Factors | Down-scaling Procedure No. | Test Set | SRResNet      | EDSR          | EWnet         |
|--------------------------|----------------------------|----------|---------------|---------------|---------------|
| x2                       | 1                          | Set5     | <b>24.714</b> | 24.681        | 24.436        |
|                          |                            | Set14    | <b>22.781</b> | 22.752        | 22.544        |
|                          | 2                          | Set5     | <b>29.383</b> | 27.807        | 29.292        |
|                          |                            | Set14    | 26.161        | 24.958        | <b>26.172</b> |
|                          | 3                          | Set5     | 23.670        | 24.294        | <b>24.666</b> |
|                          |                            | Set14    | 22.022        | 22.482        | <b>22.751</b> |
|                          | 4                          | Set5     | 22.589        | 22.461        | <b>23.475</b> |
|                          |                            | Set14    | 21.209        | 21.190        | <b>21.972</b> |
| x4                       | 5                          | Set5     | 23.323        | 23.316        | <b>23.386</b> |
|                          |                            | Set14    | 21.531        | 21.630        | <b>21.657</b> |
|                          | 6                          | Set5     | <b>23.346</b> | 23.289        | 23.034        |
|                          |                            | Set14    | 21.493        | <b>21.539</b> | 21.117        |
|                          | 7                          | Set5     | 24.814        | 24.763        | <b>24.829</b> |
|                          |                            | Set14    | 22.830        | 22.742        | <b>22.758</b> |
|                          | 8                          | Set5     | 23.943        | 21.910        | <b>24.614</b> |
|                          |                            | Set14    | 22.239        | 20.892        | <b>22.599</b> |

### 3.5 EnsemNet: Ensemble Super-resolution CNN in Wavelet domains for Small Data and Diverse Compressive Models

We have leveraged the advantage from multiple wavelet sparsifying transform domains in the study of ESnet and EWnet. The neural networks explore the feature space across different transformation domains and extract informative features as material for the final reconstruction convolutional layer. In ESnet, we use an individual network as the combiner for ensemble reconstructions from different SparseFnets. It has been proved that an ensemble can deliver more accurate results than reconstructing on a single domain. With EWnet, we simplified the ESnet into a more uniform fashion and obtained more accurate results from a more efficient architecture. However, it is unclear whether EWnet performs an ensemble or simply treats different transform representatives equally as input variables. Ensemble methods provide a more robust approach when CNNs are trained with less representative data. Transform domains could support the CNNs with multiple sparse representations of the original image data which enrich the information so that the CNNs can be sufficiently trained even using small data sets. Since we have known that both multiple domains and ensembles can improve accuracy, we would like to propose a model that carries both of these advantages.

The small-scale ablation study was conducted on ESnet in the previous Section 3.3. We attempt to improve the performance of ESnet by leveraging benchmark algorithms as the basic image processors. The deep residual blocks in benchmark algorithms are reduced to replace the SparseFnet basic image processors. However, no convincing improvement can be observed. Then, we propose an ensemble CNN in multiple representation domains called EnsemNet. While image processing of visual imagery is often done in the presence of large sets of training data, there are many important image processing problems that do not benefit from such large collections of training data. EnsemNet provides a general method of adapting benchmark algorithms into a transform-domain-ensemble fashion. In particular, we demonstrate how our proposed technique improves the stability over strong baseline techniques for visual imagery with diverse compressive models, with a focus on the ability of our approach to function even in the presence of small data sets.

The ensemble model achieves a robust optimal result for various types of compressive models from small training data sets with the help of different transform domains. Instead of using only the shallow architecture model SparseFnet in EWnet, benchmark algorithms SRResNet [12], EDSR [13] and RCAN [58] are utilized as basic image processors in EnsemNet. In this way, we can demonstrate the sufficiency of the ensemble method over single models. EnsemNet assembles multiple basic image processors in different representation do-

mains for an optimal combined result. Taking advantage of multiple representation domains, our model stands apart from previous algorithms by adapting different small training sets. The advantage of our model is demonstrated through the comparison among individual image super-resolution CNNs in single transform domains and the ensemble model. Similarly, with the study of ESnet and EWnet, it is proved that

- Our method provides a general solution for the image super-resolution problem on diverse and complex compressive models in real-world applications.
- The optimal performance is stable over different selections of small training data sets.

In addition to the case of EWnet which focuses on multiply transform domains, EnsemNet also focuses on the advantage of the ensemble. Using this method on both SparseFnet and deep benchmark models, it is proved that our method is novel because

- based upon advantageous properties of sparsifying transform domains, our ensemble model combines the results from different domains to provide a robust solution from insufficient training data sets.

### **The architecture of EnsemNet**

EWnet focus on processing the features from different transform domains more efficiently rather than combining reconstruction results from different domains. The purpose of the following algorithm is to improve the performance of ESnet while preserving the ensemble part. The ensemble algorithm, which we call EnsemNet, adapts the contents from the limited training data and then decides the best way to combine the reconstructions from different domains to provide an optimal solution. The architecture is illustrated in Figure 32. We parallel two algorithms on the space domain and the transform domain to generate two high-resolution reconstructions respectively. The algorithm then makes the ensemble in the original image domain. The ensemble method adds the two reconstructions element-wisely. Finally, with an extra convolutional layer for feature reconstruction and additional adjustment of the combination, an output is obtained. In this architecture, all the feature extraction convolutional layers use  $3 \times 3$  kernels, and all the feature reconstruction convolutional layers use  $1 \times 1$  kernel. The ensemble itself works on the combination and final feature reconstruction, it influences the training of individual algorithms by optimizing the ensemble based on different training sets and diverse compressive models. The element-wisely addition sufficiently provides a better result than simply using convolutional feature extraction from previous layers. Moreover, the performance of EnsemNet highly depends on the performances of its individual components. We try several candidates on different domains for

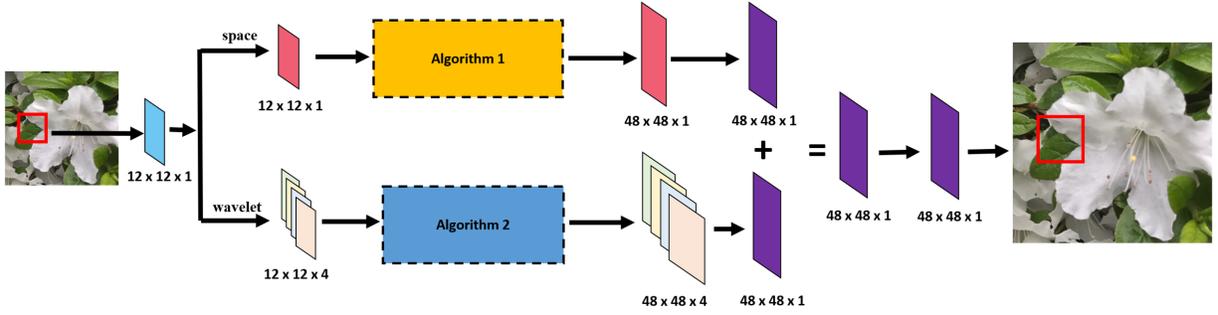


Figure 32: The architecture for the ensemble method in both the space domain and the wavelet transform domain. Two algorithms are performed separately on the spatial representation and the multi-channel wavelet representation. Then the outcomes are combined by element-wise addition. Finally, the output is refined by an extra convolutional layer.

---

**Algorithm 4** EnsemNet

---

**Input:**  $y_s \in R^{n_1 \times n_2}$

Pre-training process:  $x = \psi(y_s)$ .

Initialize  $\Phi_1$  in Algorithm 1 and  $\Phi_2$  in Algorithm 2 with randomly initialized parameters.

**While** epoch = True

**for** each iteration

$y_1 = \Phi_1(x)$ ,  $y_2 = \Phi_2(y_s)$

        Minimize  $\|\Omega - (\Phi_1(x) + \Phi_2(y_s))\|_2^2$  using back-propagation.

**end for**

**end while**

**Return**  $y = y_1 + y_2$

**Output:**  $y \in R^{m_1 \times m_2}$

---

algorithm 1 and algorithm 2 which are described in Figure 32. The best performing EnsemNet has better results than performing its individual components alone. For example, the EnsemNet with an SRResNet on the space domain as algorithm 1 and an EDSR on the wavelet domain as algorithm 2 should have improved results than SRResNet and EDSR. The best setting of EnsemNet will be found from experiments on regular visual imagery, and then applied to the MIRALON sheet application. The training pseudo-code for EnsemNet is provided in Algorithm 4.

## EnsemNet on ordinary image reconstruction

In the following, we show several experiments that were conducted to demonstrate the effectiveness of ensemble CNNs. We make comparisons among individual benchmark deep residual neural networks and ensemble CNNs over different combinations of individual algorithms. The efficiency of our best-performing EnsemNet is indicated through experiments on multiple small data sets and different measurements.

Similarly, with the experiment of EWnet, we use standard benchmark image data sets as training and testing data. Training data are from the DIV2K data set [57]. Drawn from the 800 high-resolution images, we obtain four data sets with the size of 1, 2, 3, and 5 images respectively. Testing results are compared on data sets Set5 and Set14 which consist of natural scenes. In previous studies, cubic interpolation is often chosen to be the standard compressive model. We believe it is more faithful to consider more measurements. To present the complexity and diversity of compressive models in real-world applications, four down-scaling procedures with super-resolution factors of 4 are carried out with different interpolation methods, Gaussian blur models, and Gaussian pyramid degradation, as shown in Table 11. We use packages OpenCV and scikit-image in Python to implement the measurements. A  $100 \times 100$  image would be reconstructed to  $400 \times 400$ . Each of the four down-scaling procedures consists of multiple measurements. Suppose we have a  $400 \times 400$  high-resolution image. In down-scaling procedure No. 1, the image is compressed into  $200 \times 200$  with linear interpolation first. Then, a Gaussian blur is added to smooth the image. Finally, the image is compressed into  $100 \times 100$  with the nearest interpolation.

To make full use of the limited training data, we extract small overlapping patches from the low-resolution image. The sizes for low-resolution patches are  $12 \times 12$ . Each patch is rotated in  $90^\circ$ ,  $180^\circ$ , and  $270^\circ$  to enlarge the size of the training data set. The number of training patches is 100, 150, 200, and 250 for the four data sets respectively. Another validation set of 100 patches is randomly drawn from the DIV2K data set to train the models.

Table 11: 4 compressive versions with multiple down-scaling measurements. Lanczos: Lanczos interpolation; Cubic: cubic interpolation; Nearest: nearest interpolation; Linear: linear interpolation; Gaussian: Gaussian Pyramid; GB: Gaussian Blur.

| Compressive Versions | Down-scaling Measurements           |
|----------------------|-------------------------------------|
| 1                    | Linear x2 + GB + Nearest x2         |
| 2                    | Lanczos x2 + Nearest x2 + GB        |
| 3                    | Gaussian x4 + GB                    |
| 4                    | Gaussian x2 + GB + Gaussian x2 + GB |

Table 12: Implemented models and the corresponding domains for the three settings of EnsemNet.

| <b>Names</b> | <b>Algorithm 1</b>    | <b>Algorithm 2</b>      |
|--------------|-----------------------|-------------------------|
| EnsemNet1    | EDSR - Wavelet domain | EDSR - DCT domain       |
| EnsemNet2    | RCAN - Wavelet domain | SRResNet - space domain |
| EnsemNet3    | EDSR - Wavelet domain | SRResNet - space domain |

We use the MSE loss function as our experiment suggests that other popular loss functions do not help to train models well with the sparse representatives in transform domains. Other specific settings include Adadelta optimizer [59], 32 batch-size and 1000 maximum epochs. We use a 3.20 GHz Intel core i7-8700 CPU and 64 GB memory to run the implementations.

SRResNet [12], EDSR [13] and RCAN [58] are implemented as basic models. For individual algorithms, we apply EDSR on the DCT domain, both SRResNet and EDSR on the space domain, and RCAN on the Wavelet domain. Applying the architecture in Figure 32, three combinations of ensemble algorithms are implemented on both the space domain and the Wavelet domain, as illustrated in Table 12. Packages SciPy and PyWavelets in Python are used to implement the transformations. The shape of the multi-channel wavelet representation is adjusted with zero padding.

Four experiments on two testing data sets are conducted to compare the performances for each training data set. The performance of all algorithms varies from different compressive versions and data sets. They depend on the randomly drawn small training image as well as the 'unknown' compressive models. To compare the stability of performances in different circumstances, we evaluate the algorithms with their comprehensive performances. For each training data set with each compression version, the testing PSNR (in dB) values from the seven algorithms are ranked from the highest to the lowest. Then an average ranking for each algorithm is obtained over the four training data sets. Table 13 shows the ranking result obtained from the testing PSNR values from Table 16. For example, the testing PSNR values from ENsemNet3 for compressive version 1 on Set5, rank 2, 1, 1, and 3 for training data sets of sizes 100, 150, 200, and 250 respectively. Therefore, the overall average ranking for ENsemNet3 on compressive version 1 is 1.75. The rankings are also reflected in Figure 33. For individual algorithms, EDSR on the original space domain has the most competing results on different small training sets. In two of the eight cases, SRResNet slightly outperforms EDSR. Moreover, individual algorithms have frequent gradient explosions while training with extremely small data sets, which demonstrates the instability of performance in the circumstances of this study. For ensemble algorithms, we find EnsemNet3, which

Table 13: Average rankings for testing PSNR (in dB) values. For each training data set with each compression version, the seven algorithms are ranked from the highest PSNR to the lowest PSNR. An average ranking for each algorithm is then obtained with the performances from all the four training data sets. CV: Compressive Version

| Models    | CV 1        |            | CV 2     |          | CV 3        |             | CV 4        |          |
|-----------|-------------|------------|----------|----------|-------------|-------------|-------------|----------|
|           | Set5        | Set14      | Set5     | Set14    | Set5        | Set14       | Set5        | Set14    |
| EDSR-DCT  | 7           | 7          | 5.75     | 5.75     | 6.5         | 6.5         | 7           | 6.75     |
| EDSR      | 2.5         | 2          | 2.25     | 2.75     | 3           | 3.25        | 2           | 2.25     |
| SRResNet  | 2.25        | 3.25       | 3.75     | 3.75     | 3.75        | 3.75        | 2.25        | 2.25     |
| RCAN-WVT  | 5           | 5          | 5        | 4.5      | 5.25        | 4.75        | 5.25        | 5.25     |
| EnsemNet1 | 5.75        | 5.75       | 5.25     | 5.25     | 5.25        | 5           | 5.75        | 6        |
| EnsemNet2 | 3.5         | 3.25       | 4        | 4        | 2.5         | 2.25        | 4           | 3.5      |
| EnsemNet3 | <b>1.75</b> | <b>1.5</b> | <b>2</b> | <b>2</b> | <b>1.75</b> | <b>2.25</b> | <b>1.75</b> | <b>2</b> |

combines a EDSR in the wavelet domain and a SRResNet in the space domain, has the most stable performance. This ensemble algorithm delivers a robust outstanding result in all the cases from different small training data sets on diverse and complex compressive models.

Table 14: Testing PSNR (in dB) with different models for different training data sets. The performance of all algorithms varies from different compressive versions and data sets. Table 13 shows the summary of these results. This table is provided for completeness as one can see the averages in Table 13 demonstrate that EnsemNet3 has the most stable performance on different training data sets and diverse compressive models. (Data Size: # of training patches, CV: Compressive Versions)

| Data Size | CV | Test Set | EDSR-DCT | EDSR   | SRResNet | RCAN-WVT | EnsemNet1 | EnsemNet2 | EnsemNet3 |
|-----------|----|----------|----------|--------|----------|----------|-----------|-----------|-----------|
| 100       | 1  | Set5     | 22.474   | 22.788 | 22.752   | 22.742   | 22.681    | 22.831    | 22.816    |
|           |    | Set14    | 20.975   | 21.157 | 21.149   | 21.126   | 21.114    | 21.198    | 21.190    |
|           | 2  | Set5     | 22.345   | 23.105 | 22.073   | 22.976   | 22.631    | 22.926    | 22.978    |
|           |    | Set14    | 20.869   | 21.169 | 20.633   | 21.188   | 21.009    | 21.116    | 21.207    |
|           | 3  | Set5     | 21.759   | 21.103 | 23.226   | 19.279   | 22.269    | 24.118    | 23.335    |
|           |    | Set14    | 20.976   | 20.375 | 22.099   | 20.558   | 21.541    | 22.437    | 22.043    |
|           | 4  | Set5     | 21.753   | 23.988 | 24.147   | 23.479   | 22.778    | 23.610    | 24.162    |
|           |    | Set14    | 20.586   | 22.234 | 22.331   | 21.881   | 21.368    | 21.940    | 22.317    |
| 150       | 1  | Set5     | 22.597   | 22.863 | 22.870   | 22.785   | 22.643    | 22.772    | 22.870    |
|           |    | Set14    | 21.086   | 21.225 | 21.211   | 21.172   | 21.096    | 21.168    | 21.222    |
|           | 2  | Set5     | 22.142   | 23.015 | 22.625   | 22.600   | 22.477    | 22.736    | 22.783    |
|           |    | Set14    | 20.725   | 21.158 | 20.936   | 20.982   | 20.901    | 21.043    | 21.023    |
|           | 3  | Set5     | 23.987   | 25.680 | 24.879   | 24.669   | 24.745    | 25.072    | 25.337    |
|           |    | Set14    | 22.256   | 23.352 | 22.976   | 22.627   | 22.724    | 23.107    | 23.180    |
|           | 4  | Set5     | 24.476   | 26.078 | 25.610   | 25.096   | 24.869    | 25.422    | 25.486    |
|           |    | Set14    | 22.608   | 23.693 | 23.396   | 23.025   | 22.882    | 23.335    | 23.352    |
| 200       | 1  | Set5     | 22.520   | 22.869 | 22.849   | 22.620   | 22.700    | 22.839    | 23.041    |
|           |    | Set14    | 21.042   | 21.249 | 21.233   | 21.071   | 21.118    | 21.201    | 21.373    |
|           | 2  | Set5     | 22.951   | 22.822 | 23.166   | 22.661   | 22.858    | 22.851    | 23.194    |
|           |    | Set14    | 21.150   | 20.995 | 21.188   | 20.956   | 21.096    | 20.966    | 21.199    |
|           | 3  | Set5     | 24.808   | 25.462 | 25.221   | 25.557   | 25.180    | 25.571    | 25.578    |
|           |    | Set14    | 22.718   | 23.134 | 22.968   | 23.223   | 22.994    | 23.270    | 23.223    |
|           | 4  | Set5     | 24.864   | 25.959 | 25.811   | 25.515   | 25.392    | 25.755    | 25.842    |
|           |    | Set14    | 22.798   | 23.615 | 23.416   | 23.295   | 22.793    | 23.432    | 23.456    |
| 250       | 1  | Set5     | 22.656   | 22.952 | 22.969   | 22.830   | 22.703    | 22.914    | 22.936    |
|           |    | Set14    | 21.104   | 21.304 | 21.297   | 21.206   | 21.165    | 21.297    | 21.323    |
|           | 2  | Set5     | 22.934   | 23.459 | 23.440   | 23.212   | 23.045    | 23.261    | 23.410    |
|           |    | Set14    | 21.167   | 21.464 | 21.489   | 21.304   | 21.237    | 21.321    | 21.436    |
|           | 3  | Set5     | 23.991   | 25.917 | 25.583   | 25.214   | 25.055    | 25.323    | 25.656    |
|           |    | Set14    | 22.351   | 23.507 | 23.378   | 23.083   | 22.955    | 23.261    | 23.442    |
|           | 4  | Set5     | 23.617   | 24.805 | 24.895   | 24.136   | 24.327    | 24.703    | 25.088    |
|           |    | Set14    | 21.988   | 22.843 | 22.939   | 22.426   | 22.491    | 22.845    | 23.062    |

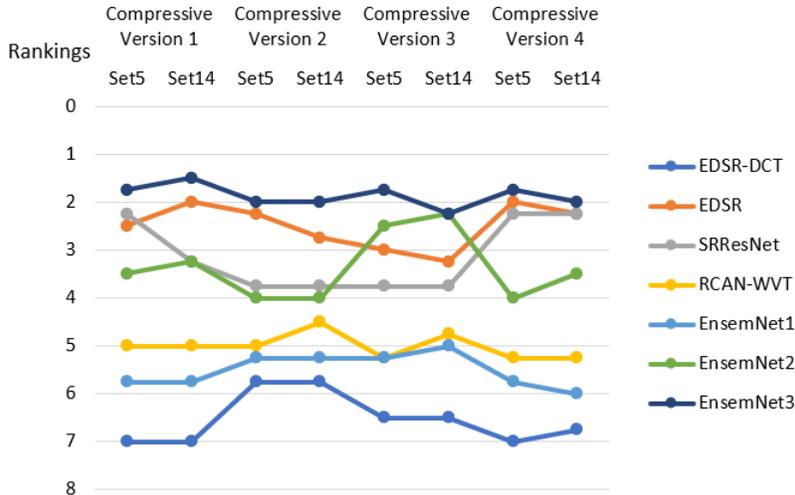


Figure 33: Average rankings of testing PSNR from the four training sets over the four down-scale procedures and the two testing data sets.

### 3.6 Voxel-based: Multi-layer Wavelet Transformations for Image Super-resolution: Applications to Voxel-based Deep Learning and Areal Density Maps of Carbon-nanotube Sheets

At this point, we have developed several algorithms to overcome the challenge of small training data in image super-resolution problems for various types of applications. We explored the feature space more sufficiently by using sparsifying domains and improved the robust performance of ordinary super-resolution models by the ensemble. However, these algorithms are not ready for the domain of MIRALON sheet areal density maps. Areal density is inherently a 3D phenomenon and the change of density as a function of depth in the sheet is critical to estimate. Herein we propose a new architecture and training approach. Through a special 3-layer voxel experiment which is specifically designed for the MIRALON problem, we can gather a limited set of training data for which the true areal density is known. This data gathering task demonstrates the need for a close collaboration between the team developing the super-resolution methodologies and the team performing the physical measurements<sup>2</sup>. Given this data set, we then develop a super-resolution architecture and training paradigm which is simultaneously efficient in its data needs as well as sensitive to density changes as a function of depth.

The architecture of EnsemNet is modified by adapting the semi-3D data. Additional data augmentation techniques are also applied to enrich the training data set. As this part

<sup>2</sup>Thanks to the Nanocomp Technologies, Inc. team for their data generation work

is studied specifically for the application of MIRALON sheet areal density maps, we will discuss the details in Section 4.2.

## 4 Application of MIRALON Areal Density Maps

MIRALON sheets are an advanced carbon nanotube-based product that provides strength, thermal and electrical conductivity, high-temperature resistance, and electromagnetic interference. Carbon nanotubes are seamless cylindrical hollow fibers. The nature of its hexagonal lattice structure and the strong bond between carbon atoms provides carbon fiber-like materials with impressive properties. MIRALON sheets are formed with extremely long carbon nanotubes that are recognized as a state-of-the-art carbon material. They provide sustainable and effective solutions to some of the toughest industry challenges involving the aerospace, energy, and electronics domains.

As described in Section 1, one of the challenges for quality control of the MIRALON sheets is to ensure the uniformity of the areal density distribution. The areal density map generated by a Mahlo QMS-12 Qualiscan Beta Transmission System (shown in Figure 5) provides a general texture of a MIRALON sheet. The emitter releases beta particles while the sensing head moves over the surface. The number of particles which pass through the sheet and reach the detector is converted into a measurement for an areal density map of the sheet. However, there is a lack of accuracy for fine details of the uneven distribution of the material in the MIRALON sheet. As areal density maps can be interpreted as images, current work in image super-resolution deep convolutional neural network (CNN) promises to reconstruct high-resolution areal density maps with fine texture details.

However, when compared to classic problems in image super-resolution, there are two challenges that need to be overcome in order to apply the fundamental theory of super-resolution to MIRALON sheets.

- This problem is a small sample problem, as only limited training data is available. We need to ensure that the model is capable of capturing the compressive model from small training data sets.
- The unknown compressive model is complex. We not only need to determine the math

behind the camera but also the physical movement of the equipment. The behavior of beta particles, the spreading distribution from the emitter head, and the mathematical conversion in the equipment are all unknown.

The measurement is clearly more complex than interpolation methods that are used in studies of regular visual imagery. Luckily, by leveraging the sparsifying transform domains and the ensemble methods into image super-resolution neural networks, we now have several candidate algorithms that could potentially provide robust accurate results on high-resolution MIRALON sheet areal density maps.

Nevertheless, there are two additional technical challenges in addition to the fundamental challenges. Before applying image super-resolution neural networks to the MIRALON sheet application, we need to understand the difference between ordinary images and MIRALON areal density maps.

- The ground-truth of the high-resolution density maps is unavailable.
- The MIRALON sheet areal density distribution has depth dependency. We need the accuracy in the thickness, but 2D super-resolution neural networks cannot provide enough information in the vertical direction.

To overcome the first technical challenge of high-resolution density map ground-truth, we need a sheet with minimal or at least known variations in the areal density. The first approach is to use a standard material called shim-stock. As introduced with Figure 7 in Section 1, shim-stocks are assumed to be perfectly uniform. Cutting the shim-stock with a template of geometric figures and letters will generate a sheet with known defect holes. In the study of SparseFnet, ESnet, EWnet, and EnsemNet, we use this version of the ground truth to train the model for the MIRALON sheet application.

However, there are still two problems with using the shim-stock as ground truth:

- The behavior of the beta particles could be different between the shim-stock and the MIRALON sheet.
- The shim-stock only has holes and areas of known density, the bias is introduced with the Gaussian blurring filter.

Therefore, we introduced another version of the ground truth with a real MIRALON sheet. In order to control the measurement error, areal density maps of small pieces of MIRALON sheet were first generated, and only the most uniform were selected to cut with the same template as the shim-stock pieces. This result in an even smaller training data set. To further study the depth dependency of the beta particles, we propose the 3-layer voxel representation

to adapt current deep CNNs to the particular application of MIRALON sheets. With the help of the proposed algorithm, a high-resolution areal density map can be reconstructed, which successfully reveals finer patterns in the MIRALON sheets.

In the following sections, we will report our MIRALON sheet application results in the order of the theory development. We will show that all the results are improved step by step as we proposed more advanced algorithms. Since EWnet and EnsemNet are both improved versions of ESnet, we will skip the MIRALON sheet application result on ESnet. The progress is divided into two sections. First, the comparison between different algorithms is illustrated with the shim-stock ground truth. Then, the MIRALON sheet ground truth is included for the study of the semi-3D voxel design. Our results demonstrate that appropriate data augmentation is essential to achieving accurate results, and transform domains could provide special information based on the image content. Using our proposed techniques, a high-resolution areal density map, rich in 3D information, can be obtained by leveraging 2D image super-resolution CNNs.

## 4.1 Super-resolution Neural Network with shim-stock ground truth

We implement the algorithm for the application of beta transmission areal density maps for MIRALON sheets. The measurement parameters of the beta transmission Mahlo system are assumed to be fixed. If the setting of the beta transmission sensor is adjusted, i.e., sensor speed, viewable range, etc., then the model needs to be trained for the new setting before applying it to the manufacturing batches. First, we test the approach on a second version of the ground truth sheet. Then, we apply the model on areal density maps of several MIRALON sheets.

### 4.1.1 Reconstruction of MIRALON Areal Density Maps with SparseFnet

Figure 34 shows the results of our models applied to the areal density map of a second piece of shim-stock with laser-cut geometric figures. White areas are materials with thick and even densities. Blue areas are cut figures with thinner densities. We attempt to increase the resolution of the areal density map with a super-resolution factor of 4. The reconstruction quality from SparseFnet is visually appealing. EWnet makes further improvements on the basis of SparseFnet. For example, outlines of the arch and the square figures in the right-bottom corner (red window) are increasingly clear from SparseFnet to EWnet. For the

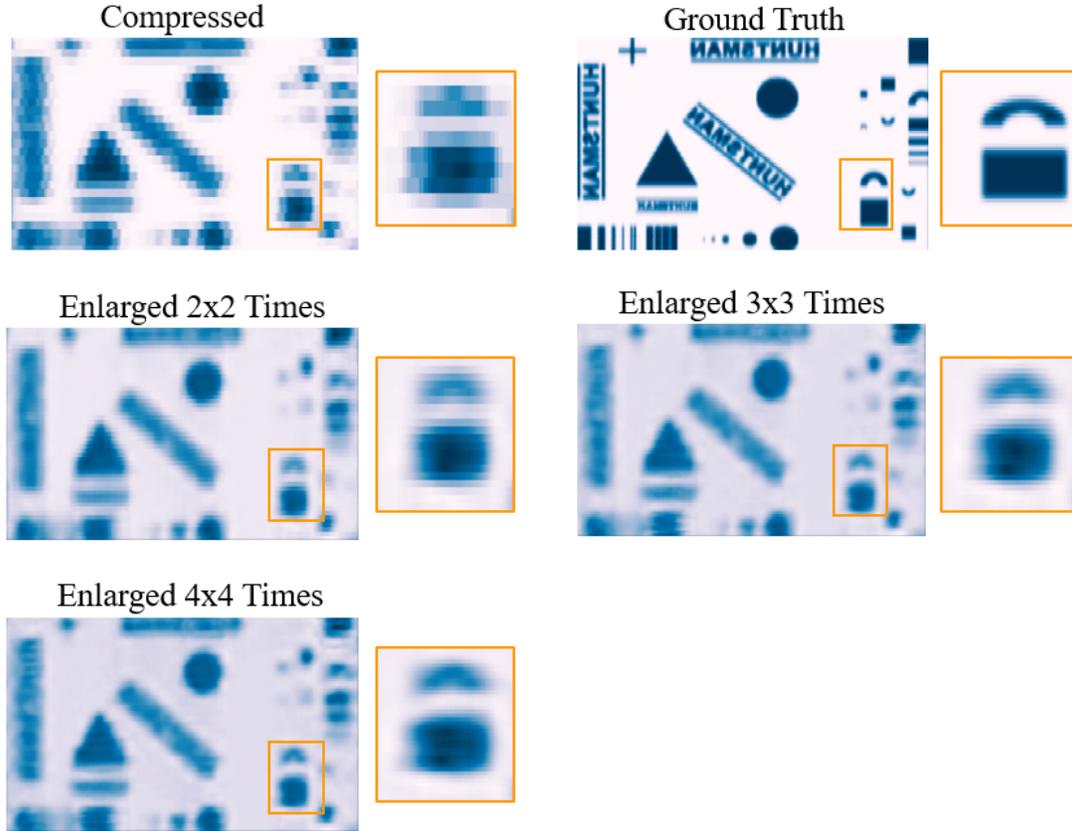


Figure 34: Reconstructed beta transmission areal density maps by the proposed algorithm for MIRALON sheets from the production line. Details for uneven density distribution are recovered.

letters on the right side (yellow window), SparseFnet reveals the outline for each letter and EWnet recovers the structure more precisely even with a clear underline.

Figure 35 shows the results of our model applied to the areal density maps of recently produced MIRALON sheets with a super-resolution factor of 4. The Mahlo beta transmission system uses the same parameter settings as for both versions of the ground truth sample. Additional details about defects and areal density variation are revealed in the high-resolution reconstructed density maps. Taking the green box section from the first sheet in Figure 35 as an example, the woven pattern running through the low-density valleys (deep blue area) and high-density hills (light blue area) is discovered more precisely. This indicates that our model is practical for the application.

A side effect can be observed from the result is the measurement error brought by the mesh screen. The mesh screen is a bottom plate underneath the scanning target on the Mahlo equipment. It is a relatively very thin layer which should not bring apparent side effects. However, the thickness of the mesh screen can be ignored with respect to the training shim-stock. When we applied the trained algorithm on the MIRALON sheet, the side effect

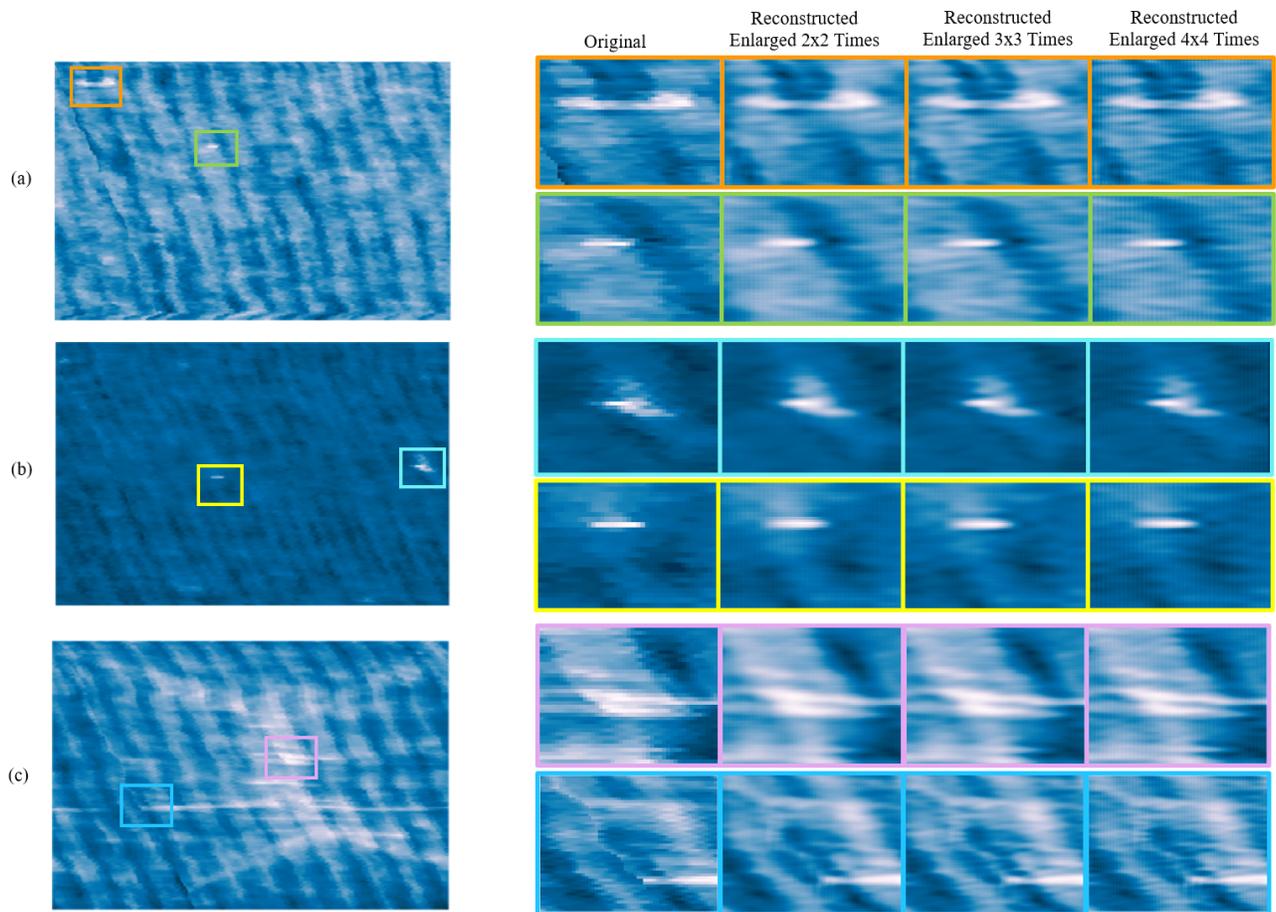


Figure 35: Reconstructed beta transmission areal density maps by SparseFnet for MIRALON sheets from the production line. Details for uneven density distribution are recovered.

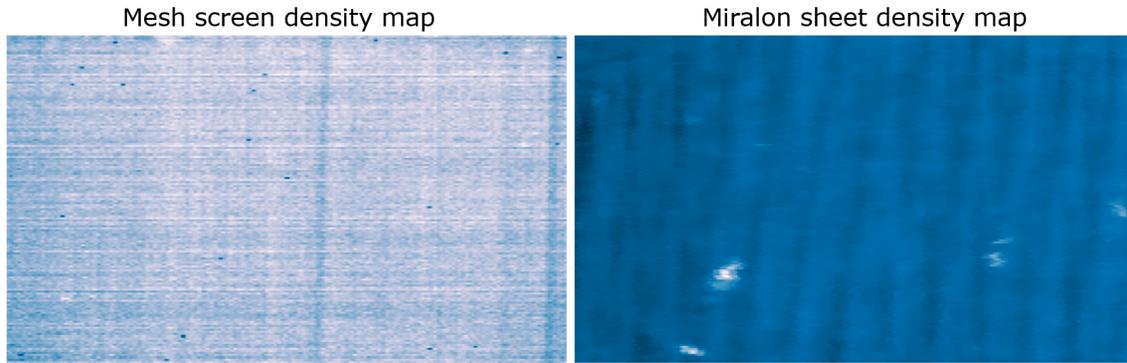


Figure 36: The low-resolution density maps of the MIRALON sheet and the underneath mesh screen.

rises because the relative thickness between the mesh screen and the scanning target sheets are different. As shown in Figure 36, though the mesh screen is very thin that can be ignored in most cases, it may shift the overall density distribution when the MIRALON sheet is thin. By this point, we make an adjustment by subtracting the mean density of the mesh from the high-resolution output as we cannot detect the mesh screen when using the shim-stock. In the upcoming Section 4.2, this problem will be solved by using MIRALON sheet as the ground truth.

### 4.1.2 Reconstruction of MIRALON Areal Density Maps with EWnet

First, we test the approach on a second version of the ground truth sheet. Then, we apply the model on areal density maps of several MIRALON sheets. Figure 37 shows the results of our model applied to the areal density map of the second piece of shim-stock with laser-cut geometric figures. White areas are materials with thick and even densities. Blue areas are cut figures with thinner densities. The reconstruction quality is visually appealing. For example, the outlines of the arch and the square figures in the right-bottom corner are increasingly clear as the enlarge factor grows. Figure 38 shows the results of our model applied to the areal density maps of three recently produced MIRALON sheets. The Mahlo beta transmission system uses the same parameter settings as for both versions of the ground truth sample. Additional details about defects and areal density variation are revealed in the high-resolution reconstructed density maps. Taking the green box section from the first sheet in Figure 38 as an example, horizontal patterns in the low-density valley (deep blue area) are discovered more precisely as the enlarge factor increases. This indicates that our model is practical for the application.

At this point, we can make the comparison between SparseFnet and EWnet. In Figure 37, the outlines of the figures are with less deformation in the reconstruction from EWnet. In particular, though SparseFnet could reconstruct some blurring letters of 'Huntsman' from the single DCT domain, it is hard to recognise the letters. On the contrary, EWnet provides a very clear image of the 'Huntsman' letters. In Figure 39, it can be observed from the SparseFnet reconstruction that there are some additional grid patterns. This may be caused by the DCT representatives. DCT representatives are very sparse, when the training image patch is not big enough and when the training data set is extremely small, it brings extra noise frequency. EWnet helps to solve this issue by considering more types of sparsifying transformations. Since EWnet has a more robust performance and can make adjustments based on different image content, a smoother reconstruction is delivered. The shape and area of the defect spots are easier to be measured. And more importantly, we have a much clearer vision of the woven lay-down patterns.

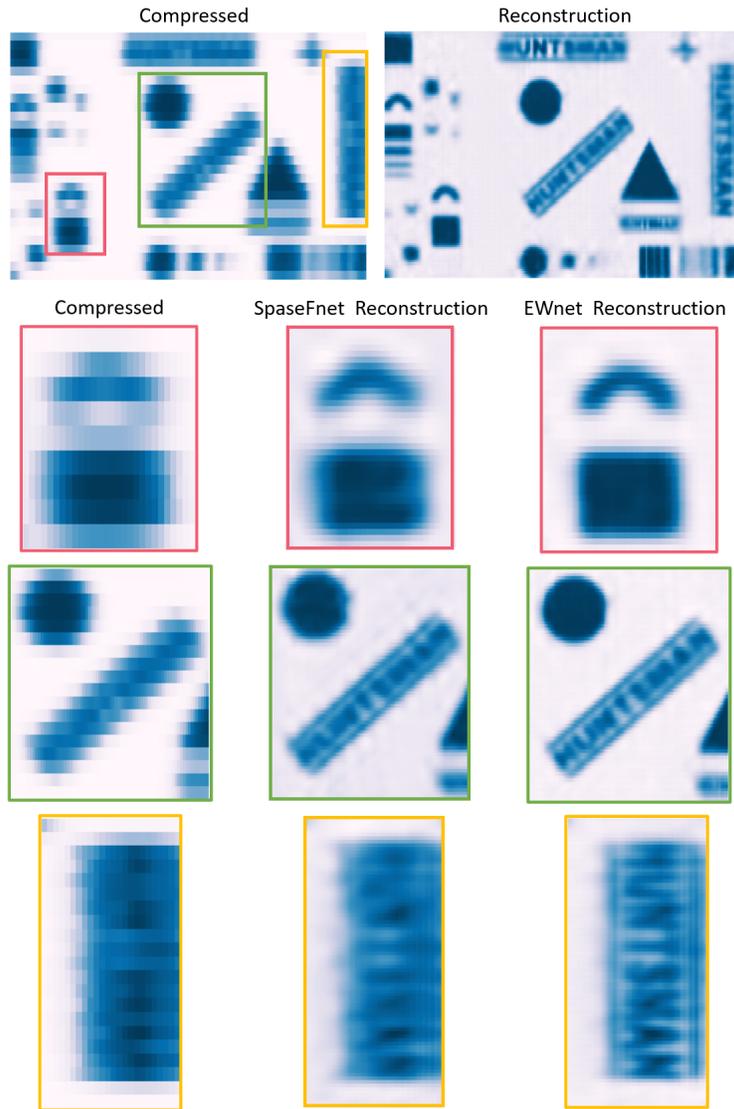


Figure 37: Areal density maps for the Patterned shim-stock. The outline of cut figures, especially the letters, is revealed more clearly in the enlarged images.

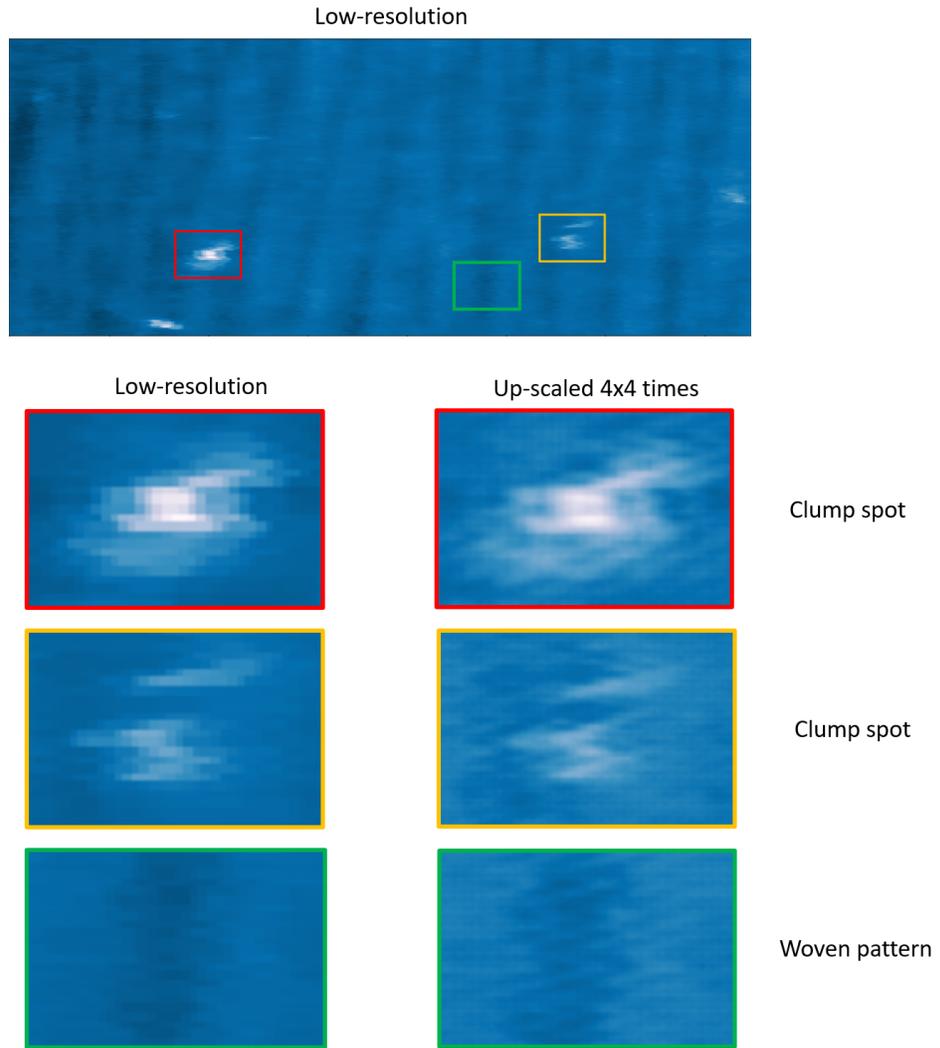


Figure 38: Reconstructed beta transmission areal density maps by EWnet for MIRALON sheets from the production line. Details for uneven density distribution are recovered.

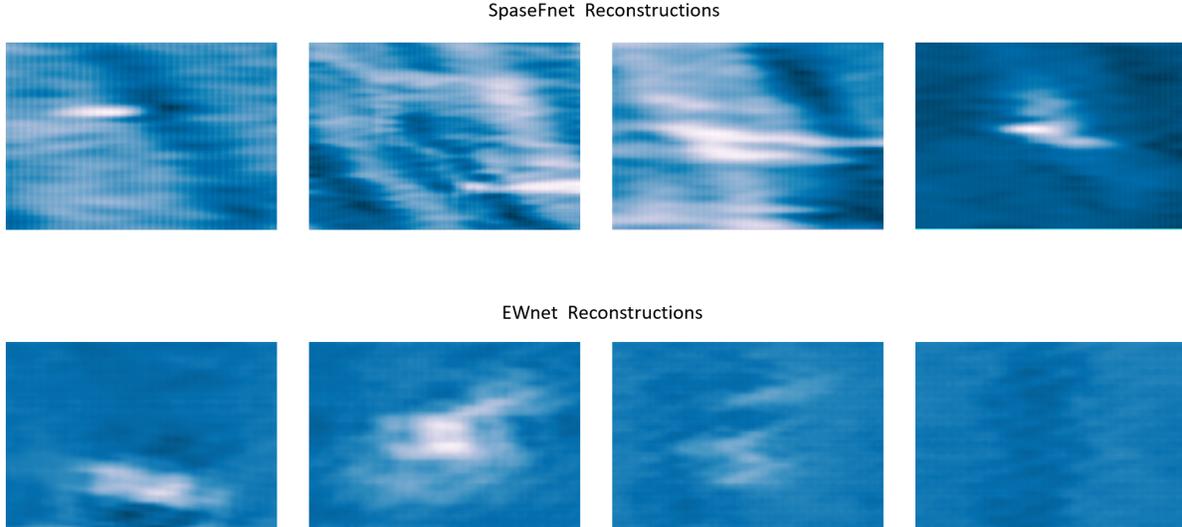


Figure 39: Comparison between SpaseFnet reconstructions and EWnet reconstructions. With the help of multiple transform domains, EWnet delivers more accurate results with less noise.

#### 4.1.3 Reconstruction of MIRALON Areal Density Maps with EnsemNet

EnsemNet focus on the sufficiency of the ensemble over different benchmark algorithms. In the study on ordinary images, we found that the version of EnsemNet3 has the most stable and robust performance on different complex compressive measurements. Here, we implement the version of EnsemNet3 to the application of beta transmission areal density maps for MIRALON sheets. A better visualization of the reconstruction are shown in Figure 40. We use the color bar to indicate different density ranges, helping to identify defects and patterns. Additional details about defects and areal density variation are revealed in the high-resolution reconstructed density maps. Taking the zoomed section from the sheet in Figure 40 as examples, the shape and the area of defective spots can be discovered more precisely from the super-resolution reconstruction. This indicates that our method is practical for the application.

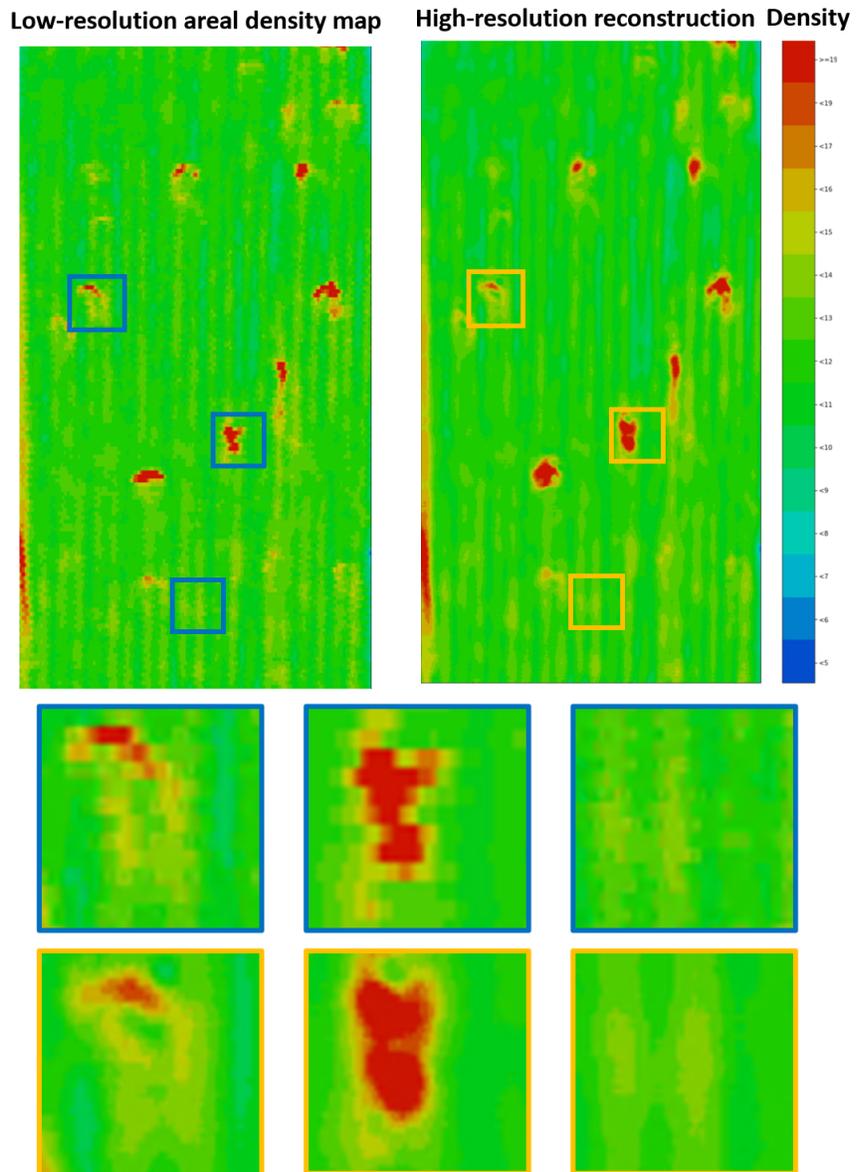


Figure 40: Reconstructed beta transmission areal density maps by EnsemNet3 for MIRALON sheets from the production line. Details for uneven density distribution are recovered.

## 4.2 Semi-3D Super-resolution with MIRALON Sheet ground truth

Unlike ordinary images for most image reconstruction problems, the beta transmission areal density map is a 2D projection of the MIRALON sheet whose density is depth-dependent. Figure 44 illustrates the measuring equipment setup. The particle emitter on the Mahlo Beta Transmission System spreads beta particles with an unknown distribution at each picture frame. As it moves along the surface at different speeds and directions regarding specific settings, the total number of particles that penetrate the sheet is converted into the camera measurement. The behavior of the beta particles varies for different sheet densities. Also, standard image super-resolution CNNs require training data for which the true density is known. We use small pieces of MIRALON sheets in which geometric figures are cut. However, such small training sets cannot meet the data requirement of most image super-resolution CNNs.

Therefore, this study overcomes three major challenges:

- First, the distribution of the beta particle spread is unknown. The particle spreading in each single picture frame is not necessarily a uniform or Gaussian distribution. The observation from the equipment is obtained from overlapping picture frames. Combined with different emitter settings (i.e., sensor speed, visible range, etc.), the camera measurement model is complex and not known *a priori*.
- Second, the penetrating power of the beta particles with regards to the different thicknesses of the material is unknown, and the paths of the particles after hitting the surface are not necessarily linear. Therefore, the number of particles received at the detector head and the thickness of the material are not necessarily in a linear relationship. Therefore, the projection of the 3D density to a 2D image is complex.
- Since the destructive measurements are expensive, only a small collection of training data is available.

To resolve the challenges, we first propose a 3-layer voxel architecture using rotated image patches for data augmentation. Then, sparsifying transform domains are utilized to enhance the density distribution patterns. We make the best use of limited data to represent the features in the specific application study. 3D voxels are made from 2D pixels to simulate the vertical thickness distribution. Voxel patches are rotated with small angles to enrich the training data set and, at the same time, avoid over-fitting. Through such practical measurements, we are able to reconstruct fine texture details from the heavily down-sampled beta transmission areal density maps of MIRALON sheets. In addition, wavelet sparsifying domains are adapted to provide more details of interests based on specific image content.

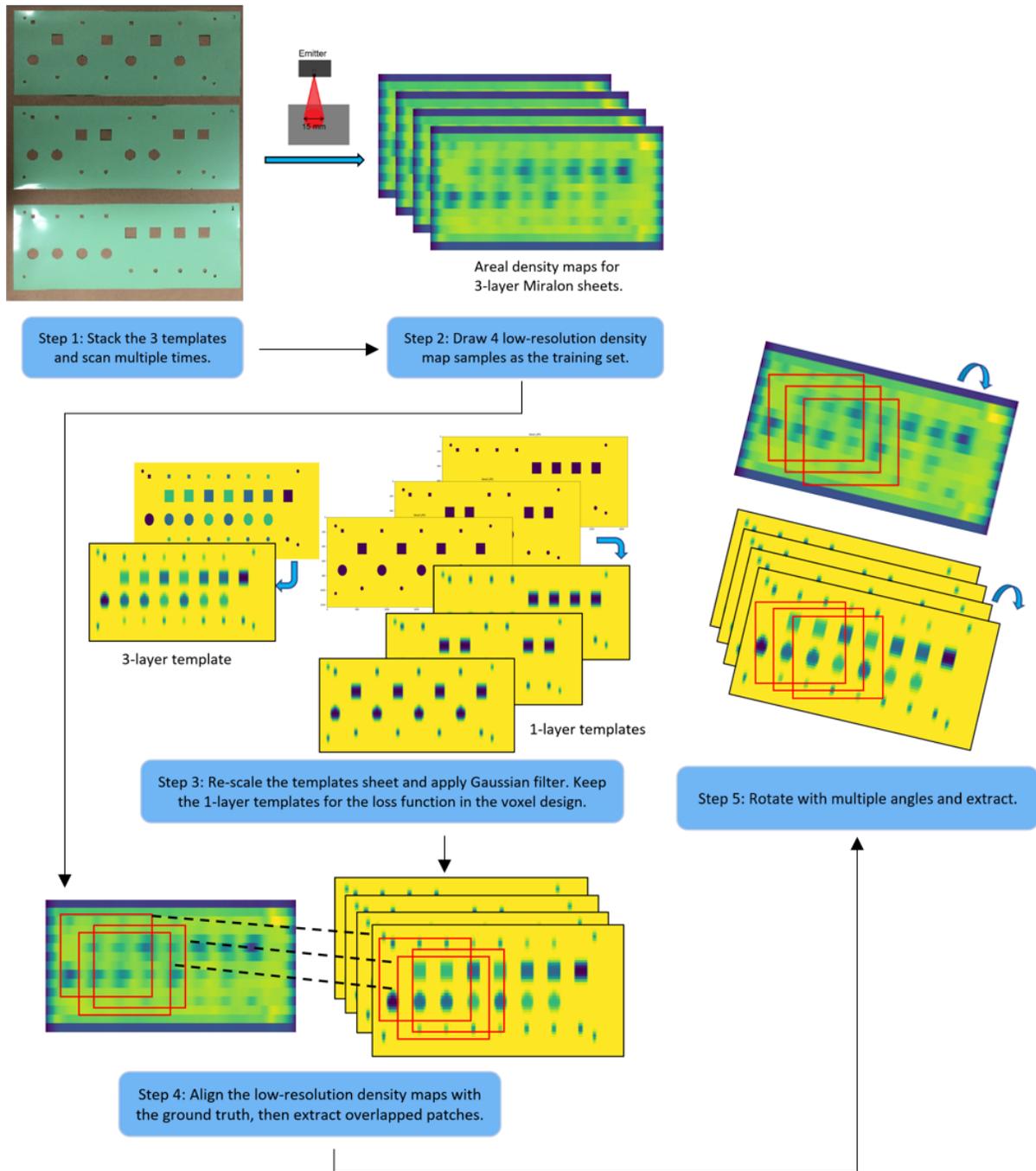


Figure 41: Flowchart of generating training data for the 3D voxel experiment design.

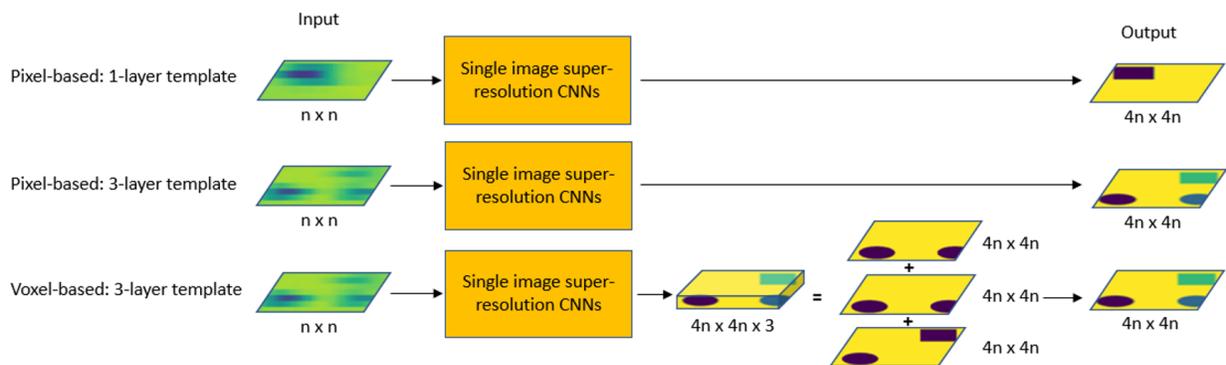


Figure 42: Different experiment settings. Both the input and the output are 2D patches in pixel-based settings. In voxel-based settings, the 2D patch is processed by single image super-resolution CNNs into an up-scaled 3D voxel patch. The high-resolution patch is then obtained by compressing the vertical axis of the voxel patch.

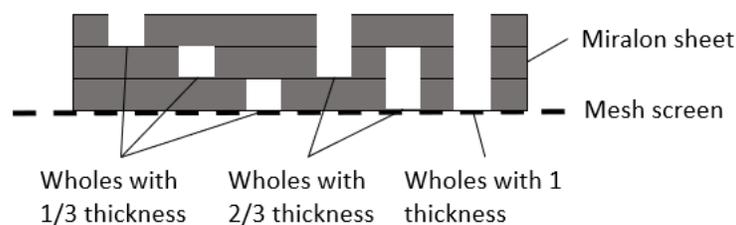


Figure 43: Cross section of the 3-layer sample. There are holes with 1/3, 2/3, and 1 thickness distributed in different vertical positions.

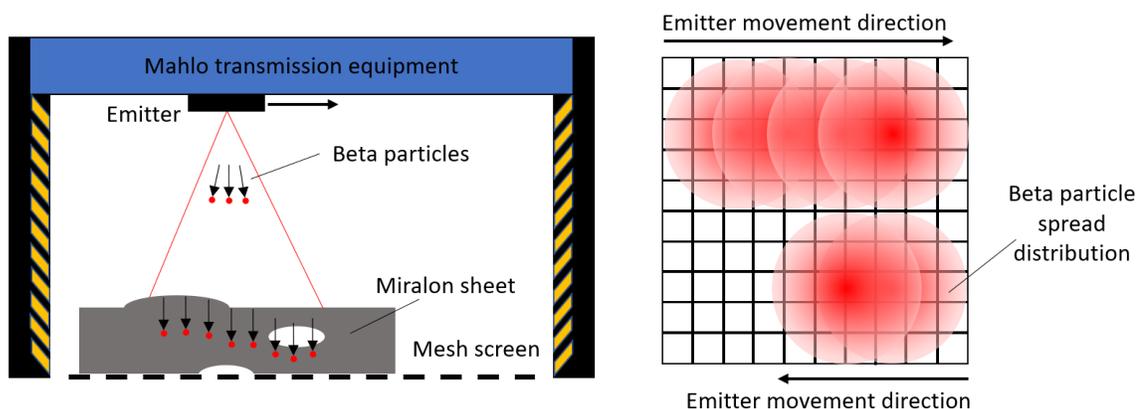


Figure 44: The MIRALON sheet is scanned on a mesh screen. As the sensing head moves, frames of the particle spread overlap with regard to the settings (i.e., sensor speed, visible range, etc.) of the equipment. The number of beta particles penetrating both the uneven MIRALON sheet and the screen is converted into the measurement.

By separating different frequencies, the wavelet transformation sensitively explores different voxel layers providing more evidence on MIRALON lay-down patterns.

#### 4.2.1 2D super resolution methods

We employ 2D super-resolution CNNs to the areal density map by treating the beta transmission measurement as an image compressive measurement. The architecture is leveraged from EnsemNet. Instead of combining 2 components in the 2D EnsemNet, we adapt the neural network with a 3-components structure for the 3D voxel design. With experiments specifically designed for this problem, SRResNet, EDSR, and EWnet are utilized to study the performance of the 3D voxel design and image patch augmentation. Then, SRResNet, EDSR, and RCAN are employed for reconstructing MIRALON sheet areal density maps from the sparsifying transform domains.

#### 4.2.2 Voxel construction

The 2D areal density map obtained from the beta transmission equipment is a projection that is missing the information in the vertical axis. The penetrating behavior of the beta particles is complex due to the different thicknesses of the material, different distribution of the defective spots, and other variations in the density. For example, as shown in Figure 44, the movement of beta particles is not necessarily linear, and may not be the same when penetrating areas of different densities. An area with an air bubble within the sheet and an area with a hole on the surface may have the same overall thickness but are shown differently on the original areal density map. In general, the behavior of beta particles is more consistent on the surface and varies more with depth. Therefore, the lack of vertical details within the sheet may lead to a misunderstanding in the sheet quality. Accordingly, we wish to recover more details from the thickness and generate a semi-3D areal density map.

As there is no ground truth in high-resolution areal density maps, the first challenge is to generate training data for super-resolution CNNs. There are few pieces of MIRALON sheets available to provide smaller areas with minimal density variation which can be assumed of uniform density distribution. We use templates to cut these small piece samples to create known density variation for studying the behavior of the beta particle measurements. Three templates with cutouts of geometric figures are designed as shown in the first step in Figure 41. Three samples are laser-cut into the shape of each template and then stacked into one

sample to obtain holes with  $1/3$ ,  $2/3$ , and  $1$  thickness of the sheet. Figure 43 shows the cross-section of the sample. The holes are distributed in different vertical positions. The sample is then scanned by the beta transmission equipment multiple times to obtain 2D low-resolution areal density maps as shown in the second step in Figure 41. The down-sampling rate is extremely low, making the geometric figures difficult to recognize in the low-resolution images.

In addition, one of the three layers is scanned separately to generate data for an ordinary 2D pixel-based experiment to compare with our proposed methods. Figure 42 interprets the algorithms. Both pixel-based 1-layer and 3-layer designs treat the density map patches as ordinary 2D images. The 1-layer design uses the data from the single-layer scan while the pixel-based 3-layer design uses the same data as the voxel design. In the voxel-based 3-layer design, the single image super-resolution CNNs process the  $n \times n$  low-resolution density patch as a regular single-channel image and output a high-resolution  $n \times n \times 3$  voxel density map patch as a multi-channel image. Each channel in the voxel patch preserves the features in the corresponding layer. A high-resolution patch is obtained by compressing the vertical axis of the voxel patch.

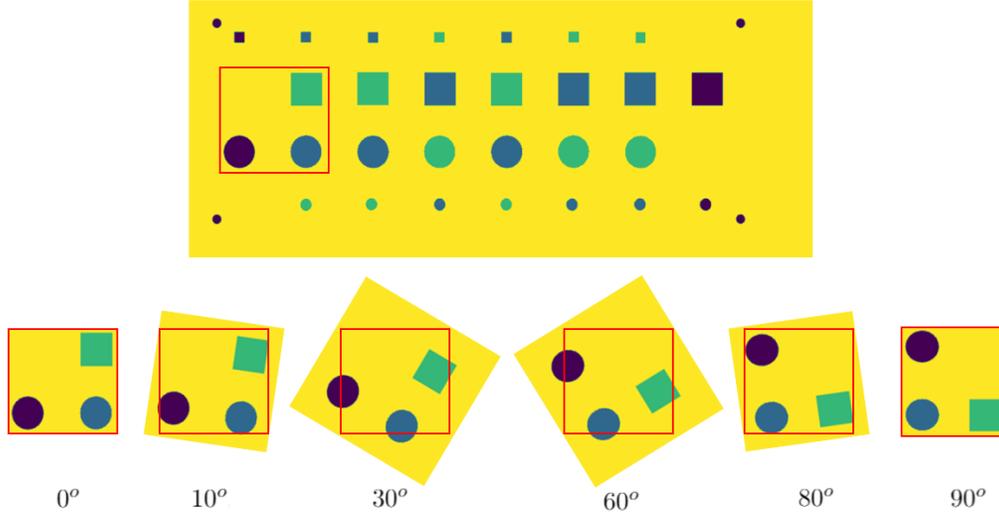


Figure 45: The sheet is rotated with multiple angles, then cropped into voxel patches to generate more representative training data avoiding over-fitting.

### 4.2.3 The rotation of voxel patches

The cutout holes in the samples are more uniform than the real MIRALON sheet defects. Therefore, they cannot be representative of the entire population of the real density variation and result in shifts or skewness in the distribution. Also, as the process of gathering real data is quite expensive, we are not able to measure a wide range of different defect types. An effective technique for improving the accuracy in image processing problems is data augmentation [60, 61]. In this study, we not only augment the data to generate more training data to avoid over-fitting but also to enrich the features of the algorithm. To help in the representation of the complex real-world defects, we rotate the voxel patches from the training data to simulate defects of different shapes. In this way, we take a small set of real-world training data and augment it to assist in the training of our super-resolution CNNs. Figure 45 provides an example of our data augmentation technique. Note, such rotations are known in other domains [62–64], but here we demonstrate their importance for 3D voxel problems in the low data limit. The flowchart for generating the training data is shown in Figure 41 with extra details.

However, the choice of wavelets is crucial as the performance depends on how well it suits the particular application [20]. Although general guidelines and suggestions for choosing the wavelet family toward specific areas are available from literature [21, 23–25], we do not have enough data to understand the distribution of the MIRALON areal density maps as the sample may not well represent the entire population. Alternately, we adapt the dropout convolution layer to avoid over-fitting and choose db6 from Daubechies wavelet family as we

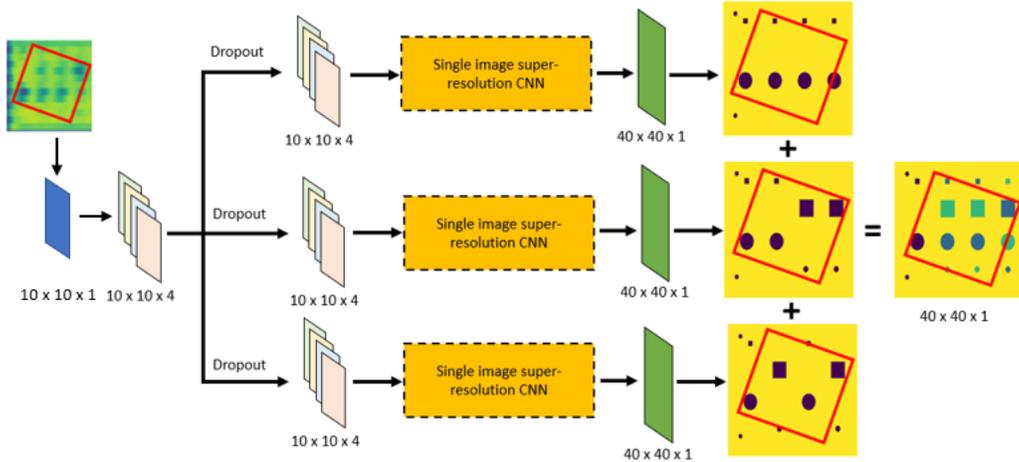


Figure 46: The architecture of the 3-layer voxel design on the wavelet domain. 10% of the information is dropped before proceeding into the individual CNNs. Each parallel CNN studies the measurement of the transmission equipment from the wavelet sparse representatives and delivers high-resolution images in the original space domain. Then, the three high-resolution outputs are compressed into a 3-layer voxel output.

observe it has an outstanding performance of revealing MIRALON lay-down patterns from the experiments. The architecture of the 3-layer voxel design on the wavelet domain is shown in Figure 46. Voxel layers are trained independently in parallel. 10% of the information is dropped before proceeding into the individual CNNs for the three layers. Each CNN studies the measurement of the transmission equipment from the wavelet sparse representatives and delivers high-resolution images in the original space domain. Notice that the loss function for individual CNNs is calculated regarding each of the corresponding 1-layer ground truths. Finally, the three high-resolution outputs are compressed into a 3-layer voxel output.

#### 4.2.4 Semi-3D Super-resolution result

The effectiveness of the proposed method has been tested with respect to numerical values. However, the testing samples are cutout pieces from the templates and therefore cannot represent the defects and distribution patterns in the real MIRALON sheet product well. We suggest using Table 16 and Table 17 as numerical references and emphasize the visual accuracy of the real MIRALON sheet products shown in the following section. The corresponding experiment settings for 'Experiment #' are reflecting in Table 15. Histograms of areal density maps approve the reliability of the proposed methods. 2D high-resolution density maps and 3D enhanced density maps provide more intuitive results on defective spots and distribution patterns respectively.

Table 15: The Four Experiment Settings

| Experiment # | Experiment Settings                      |
|--------------|--|
| 1            | Pixel-based 1-layer template             |
| 2            | Pixel-based 3-layer template             |
| 3            | Voxel-based 3-layer template (unrotated) |
| 4            | Voxel-based 3-layer template             |

Table 16: Training and testing PSNR (in dB) and SSIM with different settings regarding to different models for 4x up-scale factor. All of the models generate the highest PSNR value from the proposed design, while the pixel-based 3-layer template design gives the best SSIM in 3 out of 8 cases. Experiment #s: 1. Pixel-based 1-layer template, 2. Pixel-based 3-layer template, 3. Voxel-based 3-layer template (unrotated), 4. Voxel-based 3-layer template

| Models    | Experiment # | Training |       | Test 1        |              | Test 2        |              |
|-----------|--------------|----------|-------|---------------|--------------|---------------|--------------|
|           |              | PSNR     | SSIM  | PSNR          | SSIM         | PSNR          | SSIM         |
| SRResNet  | 1            | 20.741   | 0.664 | 21.744        | 0.729        | 19.703        | 0.712        |
|           | 2            | 24.886   | 0.823 | 20.082        | <b>0.771</b> | 22.009        | <b>0.763</b> |
|           | 3            | 22.336   | 0.550 | 17.912        | 0.410        | 17.779        | 0.406        |
|           | 4            | 18.383   | 0.534 | <b>25.509</b> | 0.655        | <b>25.365</b> | 0.645        |
| EDSR      | 1            | 13.791   | 0.685 | 18.427        | 0.691        | 18.210        | 0.685        |
|           | 2            | 23.766   | 0.727 | 18.297        | <b>0.702</b> | 20.007        | 0.691        |
|           | 3            | 22.274   | 0.751 | 20.275        | 0.674        | 21.157        | 0.679        |
|           | 4            | 21.726   | 0.723 | <b>25.759</b> | 0.700        | <b>26.781</b> | <b>0.730</b> |
| EWnet     | 1            | 19.374   | 0.473 | 22.016        | 0.671        | 22.206        | 0.673        |
|           | 2            | 21.358   | 0.679 | 21.609        | 0.680        | 22.971        | 0.683        |
|           | 3            | 24.154   | 0.801 | 9.696         | 0.591        | 9.696         | 0.591        |
|           | 4            | 26.048   | 0.978 | <b>22.481</b> | <b>0.701</b> | <b>23.074</b> | <b>0.725</b> |
| EWnet-db6 | 1            | 20.479   | 0.544 | 19.915        | 0.578        | 20.117        | 0.575        |
|           | 2            | 21.556   | 0.672 | 19.685        | 0.695        | 21.966        | 0.669        |
|           | 3            | 22.937   | 0.781 | 21.641        | 0.618        | 21.476        | 0.616        |
|           | 4            | 17.168   | 0.511 | <b>22.422</b> | <b>0.698</b> | <b>22.663</b> | <b>0.693</b> |

Table 17: The comparison of deep residual image super-resolution CNNs performing on the original space domain and the db6 wavelet transform domain. The wavelet domain shows an advantage regarding the PSNR value. Experiment #: 2. Pixel-based 3-layer template, 4. Voxel-based 3-layer template.

| Models   | Domians             | Experiment # | Training |       | Test 1        |              | Test 2        |              |
|----------|---------------------|--------------|----------|-------|---------------|--------------|---------------|--------------|
|          |                     |              | PSNR     | SSIM  | PSNR          | SSIM         | PSNR          | SSIM         |
| SRResNet | Space               | 4            | 26.472   | 0.829 | 22.082        | <b>0.784</b> | 22.611        | <b>0.788</b> |
|          |                     | 2            | 19.457   | 0.737 | 20.775        | 0.727        | 20.777        | 0.725        |
|          | Wavelet             | 4            | 20.452   | 0.756 | 20.923        | 0.734        | 24.004        | 0.750        |
|          |                     | 2            | 22.866   | 0.734 | 21.274        | 0.693        | 21.143        | 0.699        |
|          | Wavelet<br>+dropout | 4            | 21.724   | 0.756 | <b>22.906</b> | 0.738        | <b>23.600</b> | 0.745        |
|          |                     | 2            | 23.355   | 0.739 | 21.559        | 0.714        | 22.456        | 0.727        |
| ESDR     | Space               | 4            | 24.842   | 0.767 | 19.648        | 0.733        | 20.444        | 0.733        |
|          |                     | 2            | 19.646   | 0.681 | 19.014        | 0.681        | 20.009        | 0.684        |
|          | Wavelet             | 4            | 23.535   | 0.764 | <b>23.138</b> | 0.732        | <b>23.272</b> | 0.728        |
|          |                     | 2            | 22.368   | 0.694 | 19.681        | 0.688        | 19.776        | 0.679        |
|          | Wavelet<br>+dropout | 4            | 25.107   | 0.770 | 22.434        | <b>0.754</b> | 22.460        | <b>0.751</b> |
|          |                     | 2            | 20.949   | 0.688 | 20.981        | 0.701        | 21.944        | 0.703        |
| RCAN     | Space               | 4            | 20.744   | 0.683 | 20.833        | 0.696        | 21.476        | <b>0.716</b> |
|          |                     | 2            | 20.926   | 0.644 | 18.277        | 0.671        | 19.134        | 0.652        |
|          | Wavelet             | 4            | 20.473   | 0.620 | 21.157        | 0.644        | 20.712        | 0.651        |
|          |                     | 2            | 8.805    | 0.574 | 15.512        | 0.653        | 17.851        | 0.630        |
|          | Wavelet<br>+dropout | 4            | 21.745   | 0.726 | <b>21.967</b> | <b>0.703</b> | <b>23.380</b> | 0.711        |
|          |                     | 2            | 15.662   | 0.369 | 18.747        | 0.372        | 19.827        | 0.360        |

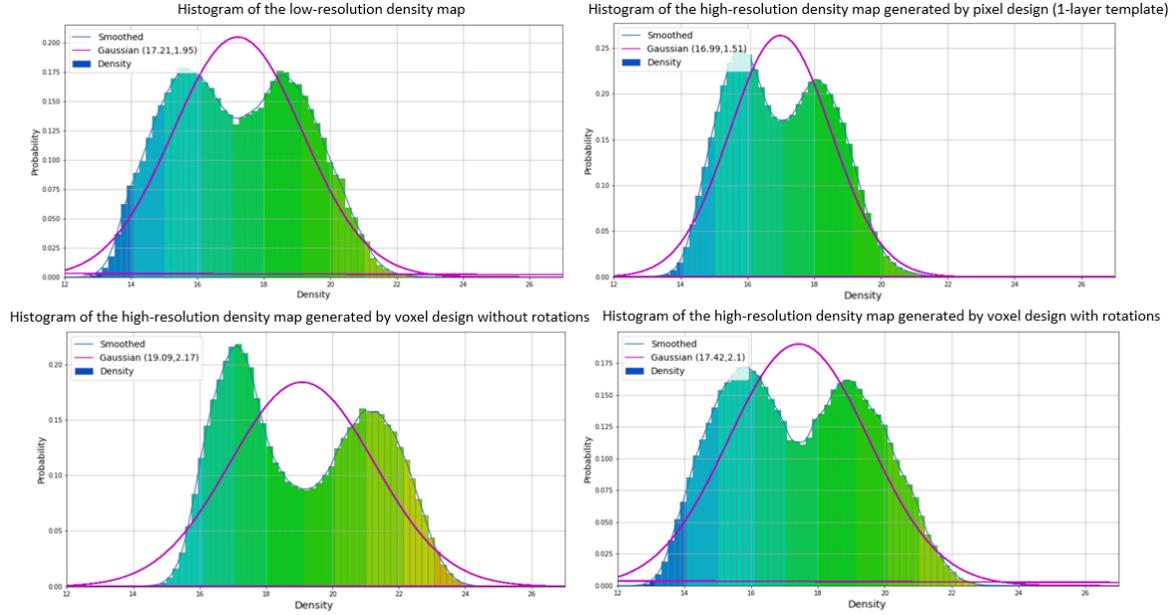


Figure 47: Histograms of the MIRALON sheet density from the original low-resolution density map and high-resolution density maps from pixel-based settings with 1-layer template design, voxel-based settings without data rotation, and voxel-based setting with data rotation. Only the voxel-based setting with data rotation well preserves the information from the original density distribution.

Ideally, the super-resolved areal density map does not make changes to the density histogram. The original low-resolution density distribution and the reconstructed high-resolution density distribution should have similar Gaussian estimations. Figure 47 illustrates how well the super-resolution model could preserve the information from the original density distribution. When the model learns only from the 1-template design, the result is over-conservative about the thin spots and the thick clumps. The density distribution is smoothed into the middle range and the Gaussian estimation has shorter tails compared with the original histogram. On the contrary, when the model learns on the 3-template voxel data without patch rotation, it over-fits the high-density area. The model focuses on the thick defect, shifting the histogram to the right. The Gaussian estimation has thick tails indicating more clumps in the sheet and brings uncertainty to the result. Despite the poor performance from the un-rotated voxel data, the rotated data corrects the performance of the voxel-based 3-layer design. With the help of patch rotation, the model delivers the closest histogram to the original data. Percentages in each range stay close to the low-resolution histogram. The Gaussian estimations have similar means and variance. Therefore, we can conclude that the proposed design super-resolves the areal density map without over-fitting.

Similar conclusions could be drawn from the 2D high-resolution density maps for a MI-

RALON sheet. Figure 48 shows closeups from a recently produced MIRALON sheet. Although the ordinary 1-layer pixel design generates a high-resolution picture, it fails to deliver texture details of interest. Thin spots and thick clumps are smoothed into the normal range. On the contrary, the voxel design without patch rotation over-emphasizes the thicker range. The performance is affected by the mesh screen which is underneath the MIRALON sheet. The result ignores the variation in the MIRALON sheet and recovers the pattern of the mesh screen. This also indicates that the multi-layer designs are able to capture different behaviors of the beta particle measurement in the vertical axis. Fortunately, the rotated data changes the performance of the voxel design. The high-resolution areal density map represents the density distribution with slight influence from the mesh screen. A clear picture of the material density distribution is reconstructed. The shape and size of defective spots are more apparent to measure.

However, the density distribution patterns are still hard to observe from the 2D high-resolution density map from the image space domain. Adapting the db6 wavelet transform domain to deep residual image super-resolution CNNs, the patterns are enhanced in 3D high-resolution density maps shown in Figure 49. From the three high-resolution density maps from the space domain, although the patterns (i.e., width, thickness, and direction of low-density valleys and high-density hills) are reconstructed more clearly compared with the original low-resolution density maps, they are still hard to observe in these visualizations. The results from the wavelet transform domain enhance the patterns. It brings more contrast to the high-resolution density maps, delivering an intuitive vision of the overall density lay-down patterns in the entire sheet.

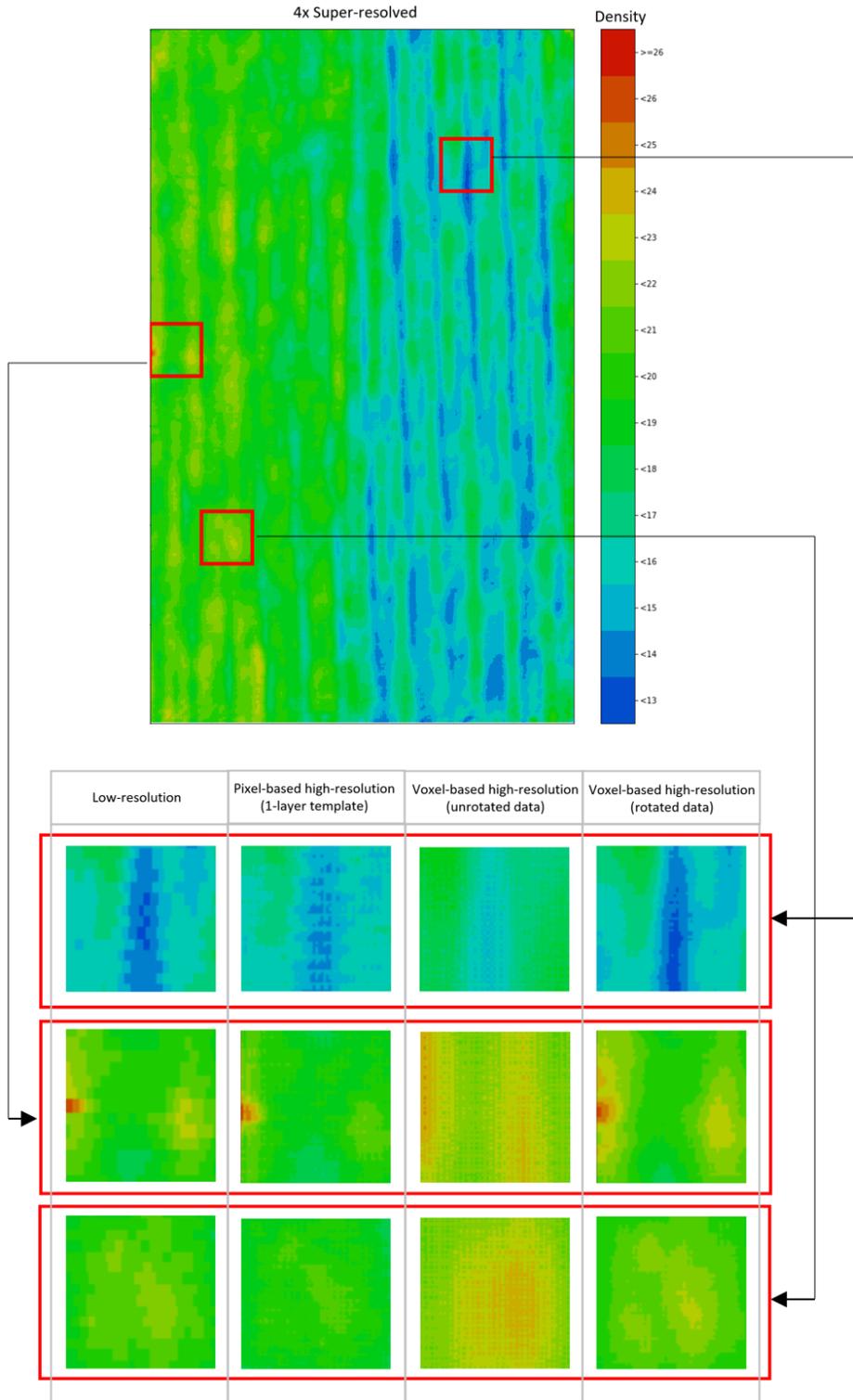
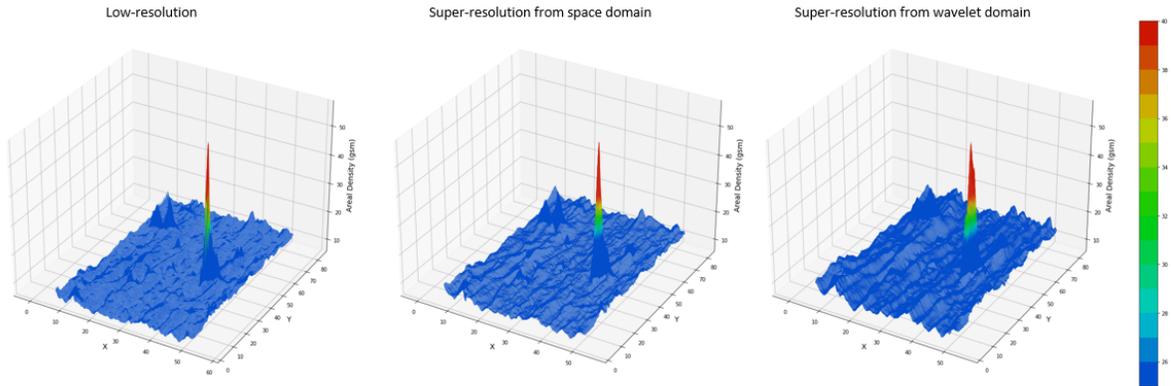
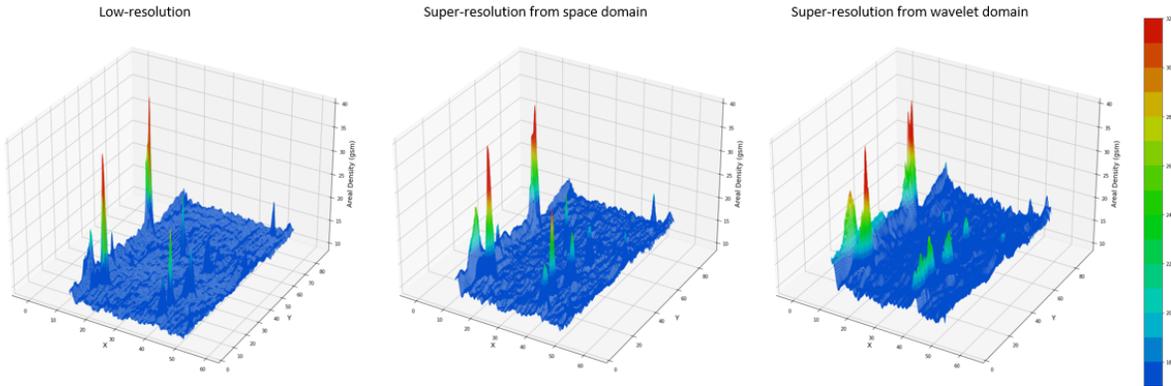


Figure 48: Super-resolution areal density map of a recently produced MIRALON sheet. Pixel-based high-resolution design underestimates both the low density and the high-density areas, and Voxel-based high-resolution design without data rotation overestimates the high-density area. Voxel-based high-resolution design with data rotation generates the best result.

Miralon Sheet No. 1



Miralon Sheet No. 2



Miralon Sheet No. 3

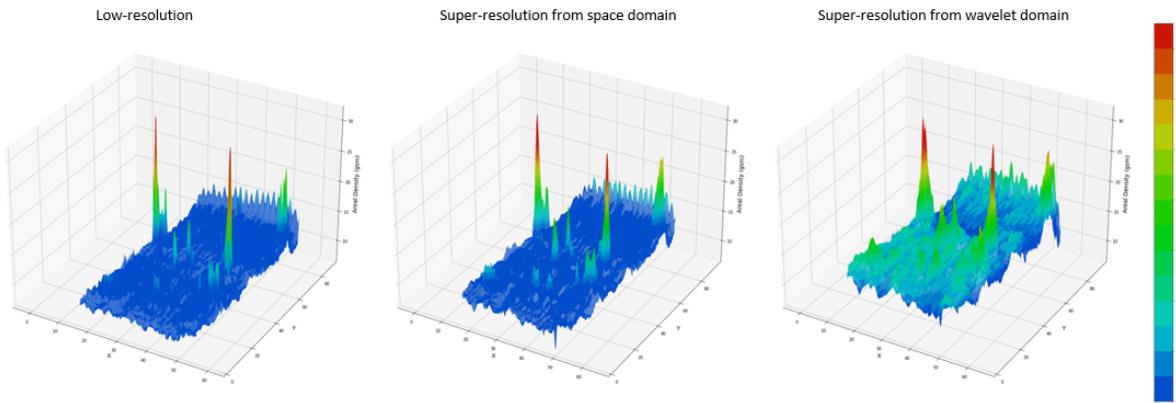


Figure 49: 3D high-resolution density maps for three MIRALON sheets. The wavelet transform domain helps to enhance the density lay-down patterns (i.e., width, thickness, and direction of low-density valleys and high-density hills), bringing a more intuitive visualization.

## 5 Future Work

Described in Section 4, though resulting in an even smaller training data set, we need to introduce the small pieces MIRALON sheet ground truth to replace the shim-stock ground truth because there are two problems with the shim-stock: (1) The behavior of beta particles is different regarding the shim-stocks and the MIRALON sheet. (2) The shim-stock has only holes and a dense area, the bias is introduced with the Gaussian blurring filter. In the experiment result from the voxel-based method, we find that the replacement largely improves the quality of the reconstruction as it solves the problem of density shifting from the shim-stock. Therefore, we can draw a conclusion that a better-measured ground truth will improve the accuracy despite the fact that it may bring larger measurement errors when generating the training data.

We plan to extend the idea from the voxel-based method and upgrade the algorithm with improved ground truth. The original method of measuring MIRALON sheet areal density distribution without the Mahlo equipment is to cut the sample into small specimens and weigh them on an analytical or ultra-micro balance. This destructive process is very expensive but could provide much more accurate views of the areal density distribution than the Mahlo method. The next step of the MIRALON application is to improve the quality of the areal density maps with the help of better ground truth from the cutting method. The size of the cut sheet is preferred to be as small as possible due to the limitation of the cost. The plan for cutting the sample sheet is given in Figure 50. Several  $20\text{cm} \times 20\text{cm}$  patches with desired density variation features (clumps and holes) will be cropped from a MIRALON sheet. One of the patches will be 'cut-and-weigh' into small specimens. Prior to the cut-and-weigh process, it will be scanned by the Mahlo equipment in a 3-layer fashion with 2 other randomly chosen patches. On the top, in the middle, or on the bottom, there will be 6 layups for each 3-piece combination. With a better ground truth on 'the cutting patch', we can conduct experiments with an inverse voxel-based algorithm. Illustrated in Figure 51, by cutting only one patch, we will have high-resolution ground truth of one of the 3 layers. In order to recover the density variation in depth, three algorithms need to be trained individually. Then we can assemble the reconstruction of every layer to obtain a voxel-based reconstruction.

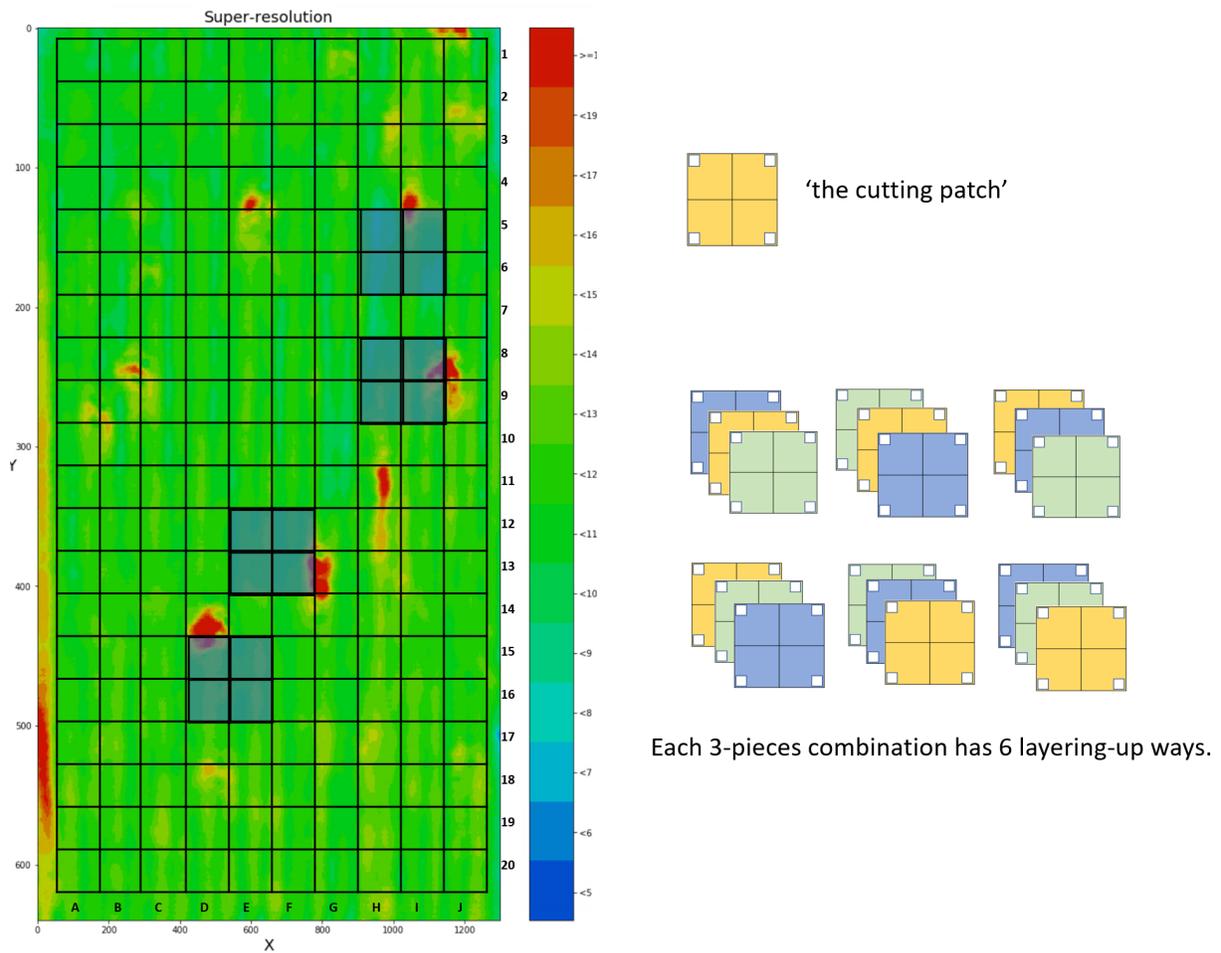


Figure 50: The template for scanning the 3-layer sheets.

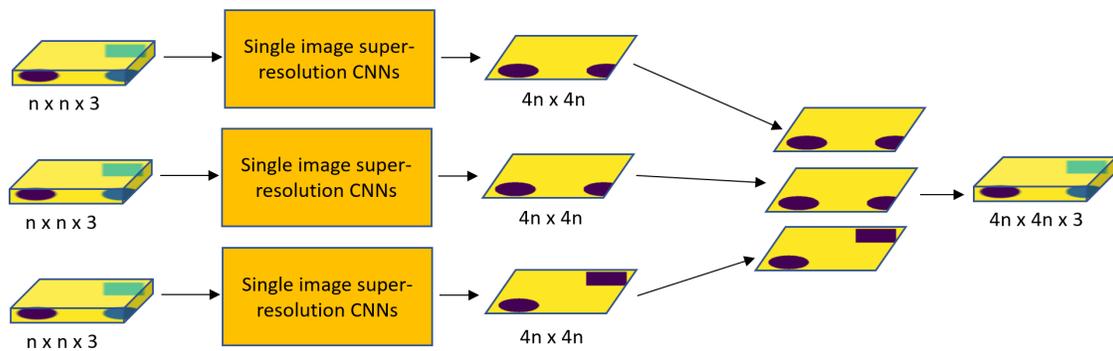


Figure 51: The inverse voxel-based algorithm for 'the cutting patch' ground truth data. Three algorithms need to be trained individually to assemble a voxel-based reconstruction.

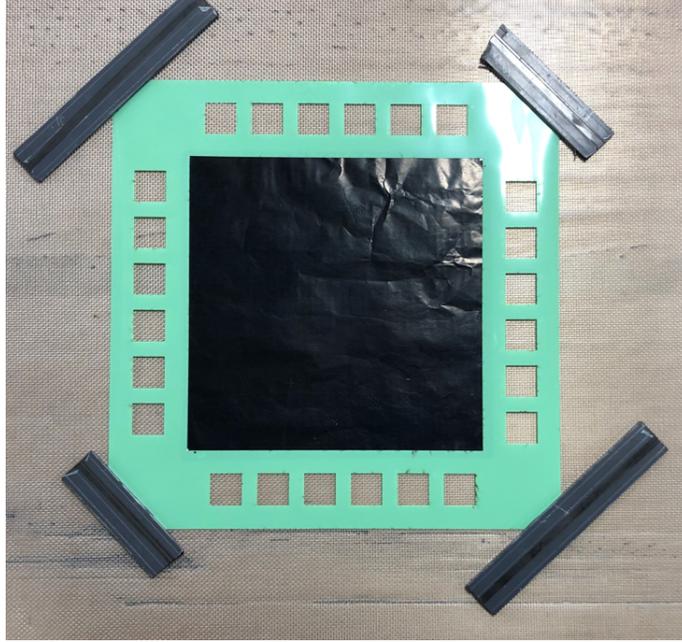


Figure 52: The template for scanning the 3-layer sheets.

One of the challenges is to align the 3 layer patches perfectly in each scan. As shown in Figure 52, a template is designed to ensure minimal displacement between different scans. An example of the Mahlo density map of the small patches is given in Figure ?? . An issue that draws attention is the rise of the shape of the pixel. From the zoom-in view of single pixels, we find the pixel from Mahlo equipment of long rectangles, which make it very hard to cut into smaller specimens. The size of the low-resolution pixel is  $1.7327cm \times 0.345cm$ . An experiment of super-resolution factor of 4 will requires high-resolution pixels of size  $0.432cm \times 0.086cm$ . Such small specimens will not only result in high cost but also brings large measurement error. Fortunately, we find that the resolution in the vertical direction and the horizontal direction on the low-resolution density maps have a large relative ratio. In other words, the limitation of the resolution is primarily brought by the vertical resolution. If we can conduct a re-resolution experiment that increases the accuracy in the vertical direction, a better view of the density distribution will be obtained. Figure 54 shows the plan of the re-resolution experiment. A  $1 \times 4$  pixel sample can be cut into 4 specimens of size  $0.86cm \times 0.69cm$ . The next plan of this research is to provide better visualization of the MIRALON sheet density map by extending the voxel-based image super-resolution algorithm reversely from the previous study with the 'cut-and-weigh' ground truth.

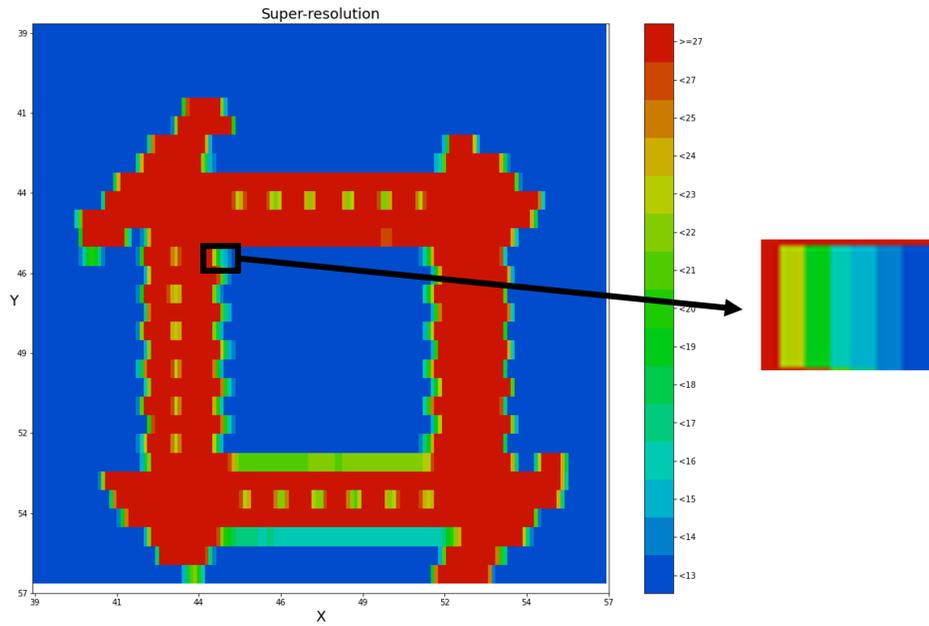


Figure 53: The density map of the cropped MIRALON sheet patch with the template.

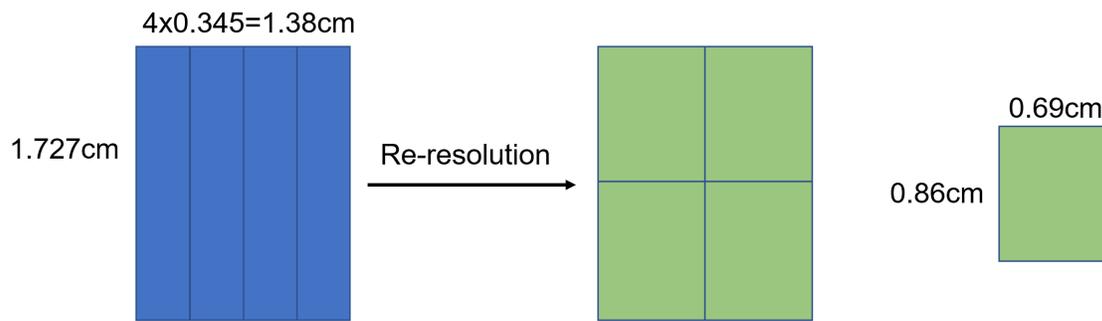


Figure 54: The size of the pixel is too small to control the measurement error of cutting the specimen. Re-resolution pixels can also provide better visualization of the MIRALON sheet density map.

## 6 Conclusion

In this four-year research on small data image super-resolution problem, we started by combining the idea of compressive sensing technique with single image super-resolution neural networks to a complex real-world application on MIRALON sheet areal density map reconstruction. First, we improved scalable single image super-resolution neural networks in the Fourier domain. It is demonstrated that with the help of the sparsifying transform domain, a shallow architecture can deliver more stable performance when the amount of training data is limited. Second, multiple wavelet domains are considered. It is found that the best-suited sparse domain differs from image contents, camera measurements, and sub-areas on an image. We proposed ensemble algorithms that leverage the advantages from multiple transform domains across different sub-image patches for general solutions for various types of images. Next, we increased the robust performance of the ensemble algorithm from the aspect of sparse feature extraction and the aspect of improved ensemble respectively. The advantages of using our proposed algorithms are illustrated through experiments on both regular images and the MIRALON sheet areal density maps for different complex compressive measurements. The performances improved as we upgrade the method step-by-step. Ultimately, we can provide a robust and efficient solution for the limitation of small training data sets in image super-resolution problems. Moreover, we extended the 2D neural networks into a semi-3D design for the MIRALON sheet application problem. With the help of the 3-layer voxel experiment design, data augmentation, and the sparsifying domains, 3D details are revealed with 2D single image super-resolution CNNs. The limitations of small training data sets and the unknown knowledge about the depth-dependent measurements from the transmission equipment are broken through. We successfully recover the fine texture details in the material product which would further contribute to the quality control progress.

# Appendices

## Automated Data Processing for Process Improvement

We have successfully recovered the fine texture details in MIRALON sheets with our algorithms. For contributing to the quality control progress, high-resolution density maps are generated for the MIRALON sheets from daily production. For providing the high-resolution reconstruction of a single MIRALON sheet, we use the Python package ipywidgets to deliver a user interface as shown in Figure 55. There are a few options including choosing the target file, initializing the density of the mesh screen, and so on. The high-resolution density map can be automatically generated along with brief analytic reports. As shown in Figure 56, the analytic report for a MIRALON sheet is composed of five parts. Both the high-resolution and the original low-resolution contour plots are generated with the color bar to provide a visualization of the density variations. The histogram of the density, the fitted Gaussian distribution, and a box plot indicates whether there are evident outliers or significant anomalies in the entire sheet. Next, we list three tables of basic statistics as a mathematical reference for the density distribution. Then, as the stripping patterns are parallel with the Y axis, a "1D" scatter plot provides the density variation in the thickness direction along with the X axis. Finally, a 3D contour plot is generated to provide more visualization details.

Furthermore, instead of generating the high-resolution reports for every MIRALON sheet manually, we would like to simplify this process into a batch operation. We write a Python

Choose the file from current directory:

File name:  ▼

Input mesh screen density here:

Input x-axis index (mm) here:

Input y-axis index (inch) here:

Generating HR 3D density map?  Yes  
 No

Cut the frame?  Yes  
 No

Figure 55: The Python user interface for generating high-resolution MIRALON density maps.

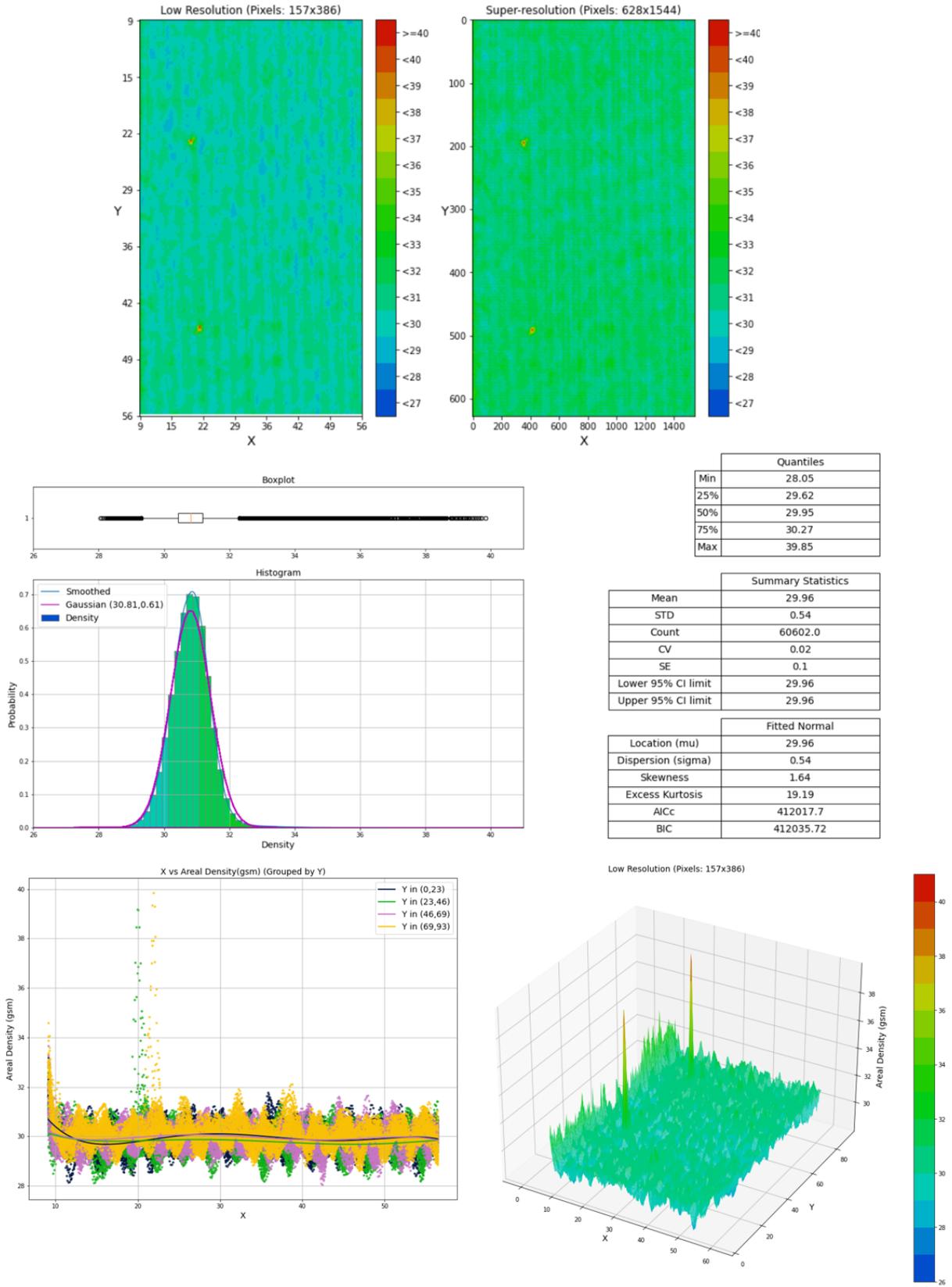


Figure 56: The analytic report of a MIRALON sheet.

script for generating the reports for the entire batch from a production day. Due to the requirement of security, this batch-operation needs to be conducted across, WPI local PC, Nanocomp virtual desktop, Mahlo drive, SQL database on Azure, and Nanocomp local PC. Figure 57 gives the flow chart of this process. First, the raw data of the low-resolution density maps which are directly from the Mahlo transmission machine are saved in Mahlo shared drive. On the virtual desktop, we could access the data from the daily batch. From there, the Python script generates the high-resolution density maps and the corresponding analytic reports for the entire batch. Instantly, the analytic reports and the high-resolution density data file are saved on the Mahlo shared drive and the SQL database on Azure respectively as the script runs on each sheet. Finally, both the WPI and the Nanocomp local ends could access the processed files.

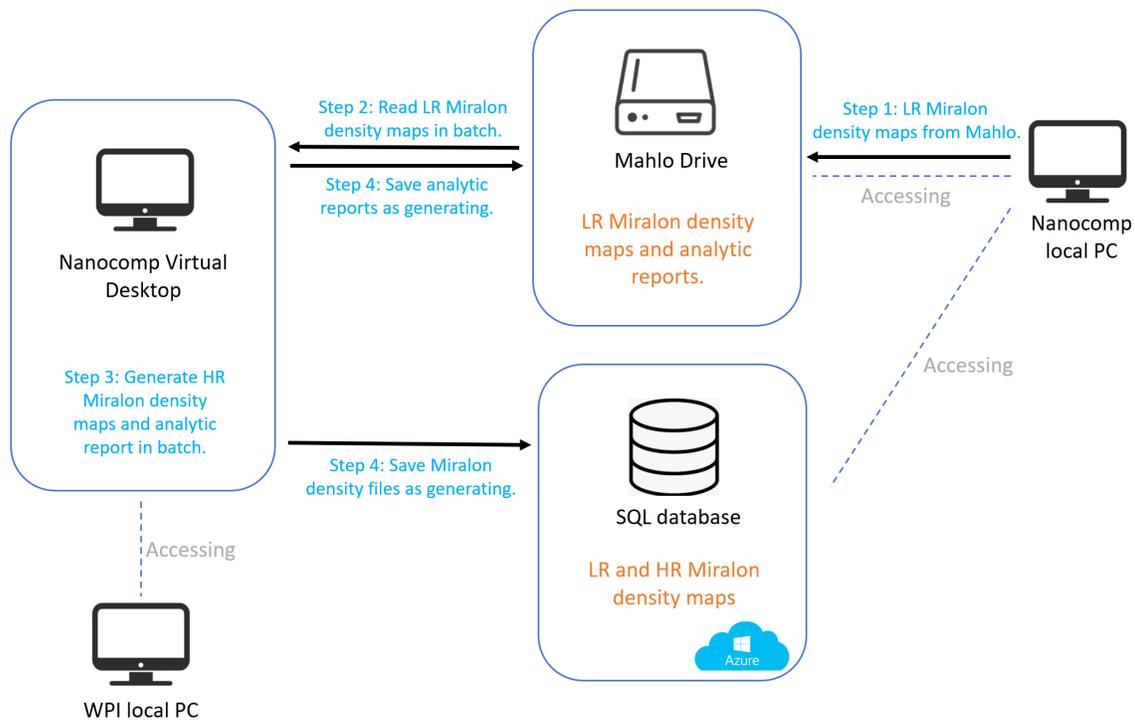


Figure 57: The flow chart of processing MIRALON density maps in batch.

## Supervised Learning for Automated MIRALON sheet evaluation

The quality of a MIRALON sheet primarily depends on its areal density uniformity. Each sheet from the daily batch production will be evaluated as 'accepted', 'conditional accepted', or 'rejected'. Figure 58 gives three sample sheets. With only a few clumps that are pointed by black boxes, the left sheet is categorized as an accepted product. The sheets in the middle and on the right are evaluated to be rejected because there are large areas of uneven density patterns which are indicated by red boxes. Currently, the decision on features for rejection is evaluated manually. We would like to use machine learning algorithms to make this process more efficient.

The nature of this problem is an image classification problem. However, from the available 400 sheets that can be studied, there might not be enough features to train the model. Especially, the number of rejected sheets is less than the number of accepted sheets which makes the training samples even less representative. Therefore, data pre-processing and feature engineering is decisive for this problem. As the rejections primarily depend on areal uneven density patterns, we propose to use 1D and 2D areal mean density as features for the algorithms. Figure 59 illustrates the methods of generating the features. The 1D features are the mean density within nine intervals on the X axis. Dividing the sheet equally into 300 small patches, the 2D features are the mean density within every patch. Feature engineering

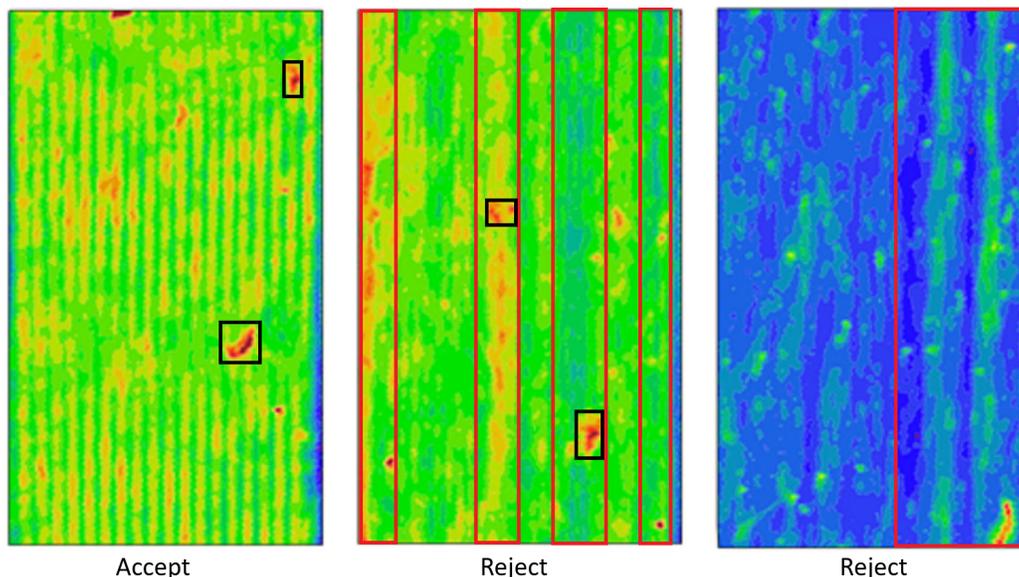
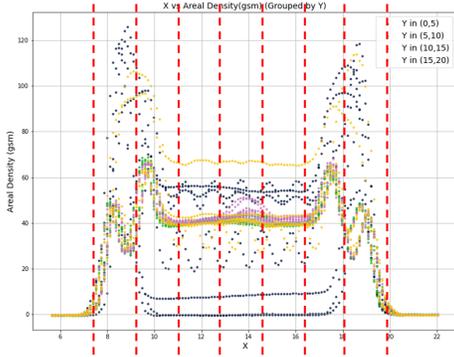
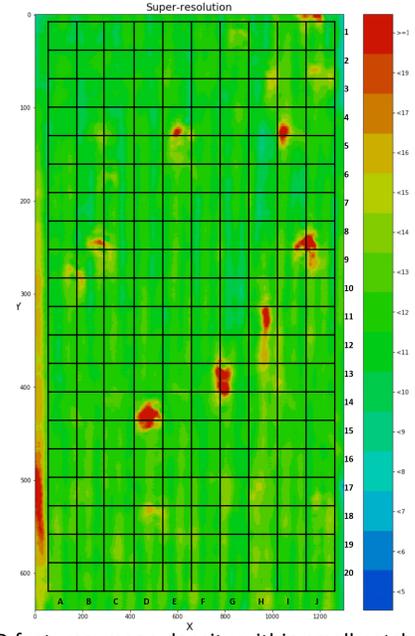


Figure 58: Sample accepted and rejected MIRALON sheets. Black boxes indicate defective spots which are not necessarily the reason for rejections. Red boxes indicate uneven areas that are the causes of rejections.



1D feature: mean density within intervals on the X axis.



2D features: mean density within small patches

Figure 59: Use 1D and 2D mean density values as features for the evaluation model. The 1D features are the mean density within nine intervals on the X axis. The 2D features are the mean density within each of the 300 patches.

turns this problem from an image classification problem to a regular classification problem.

We use a random forest model and a simple image classification CNN as approaches. The random forest has the basic settings of 100 random trees, Gini index criterion, and the maximum number of features to be used in a tree equals  $\sqrt{N}$ , where  $N$  is the total number of input features. For the deep learning approach, a simple dense network is trained. We use a fully connected layer to extract 64 features from the data, then randomly drop out 30% of the features and output the labeled class with another dense layer. Both models are trained in three settings. 1D and 2D features are used individually in the first two settings and then combined in the third set. In each setting, two cases of the original mean features and the DCT transformations of the mean features are included. The experiments are conducted on a binary "accept-reject" problem, and an "accept-conditional accept-reject" 3-classes problem. For the binary 2-classes problem, all conditional accepts are treated as rejections. The results are shown in Table 18. We focus on the binary problem as the results are more accurate. In the space domain, adding more features helps with the performances of both algorithms. Using 1D and 2D features, both the random forest and the neural network have an accuracy of around 80%. Surprisingly, the performances increase significantly to above 90% in the DCT transform domain. The DCT representatives of the areal density means are sufficient.

Table 18: The accuracy in percentages from different models with 1D and 2D areal mean density features.

| <b>Features</b>          | <b>1D</b> |      | <b>2D</b> |      | <b>1D + 2D</b> |      |
|--------------------------|-----------|------|-----------|------|----------------|------|
|                          | space     | DCT  | space     | DCT  | space          | DCT  |
| 2-classes random forest  | 78.6      | 92.9 | 75.5      | 93.3 | 80.5           | 83.3 |
| 2-classes neural network | 78.6      | 92.1 | 83.7      | 84.8 | 83.7           | 75.7 |
| 3-classes random forest  | 79.7      | 83.2 | 66.9      | 83.2 | 63.6           | 80.7 |
| 3-classes neural network | 84.7      | 84.8 | 62.7      | 80.6 | 76.3           | 79.8 |

Adding more features results in information redundancy which causes accuracy decreases for both the algorithms. At this point, we have not studied this evaluation problem intensively. In the future, this question could be fully developed into an independent study.

## References

- [1] G. Strang, “Wavelet transforms versus fourier transforms,” *Bulletin of the American Mathematical Society*, vol. 28, no. 2, pp. 288–305, 1993.
- [2] A. H. Tewfik, D. Sinha, and P. Jorgensen, “On the optimal choice of a wavelet for signal representation,” *IEEE Transactions on information theory*, vol. 38, no. 2, pp. 747–765, 1992.
- [3] E. J. Candès, J. Romberg, and T. Tao, “Robust uncertainty principles: Exact signal reconstruction from highly incomplete frequency information,” *IEEE Transactions on information theory*, vol. 52, no. 2, pp. 489–509, 2006.
- [4] D. L. Donoho, “Compressed sensing,” *IEEE Transactions on information theory*, vol. 52, no. 4, pp. 1289–1306, 2006.
- [5] H. C. Burger, C. J. Schuler, and S. Harmeling, “Image denoising: Can plain neural networks compete with bm3d?,” in *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on*, pp. 2392–2399, IEEE, 2012.
- [6] X. Mao, C. Shen, and Y.-B. Yang, “Image restoration using very deep convolutional encoder-decoder networks with symmetric skip connections,” in *Advances in neural information processing systems*, pp. 2802–2810, 2016.
- [7] K. Zhang, W. Zuo, Y. Chen, D. Meng, and L. Zhang, “Beyond a gaussian denoiser: Residual learning of deep cnn for image denoising,” *IEEE Transactions on Image Processing*, vol. 26, no. 7, pp. 3142–3155, 2017.
- [8] C. Dong, C. C. Loy, K. He, and X. Tang, “Learning a deep convolutional network for image super-resolution,” in *European conference on computer vision*, pp. 184–199, Springer, 2014.
- [9] Z. Cui, H. Chang, S. Shan, B. Zhong, and X. Chen, “Deep network cascade for image super-resolution,” in *European Conference on Computer Vision*, pp. 49–64, Springer, 2014.
- [10] C. Dong, C. C. Loy, K. He, and X. Tang, “Image super-resolution using deep convolutional networks,” *IEEE transactions on pattern analysis and machine intelligence*, vol. 38, no. 2, pp. 295–307, 2016.
- [11] Z. Wang, D. Liu, J. Yang, W. Han, and T. Huang, “Deep networks for image super-resolution with sparse prior,” in *Proceedings of the IEEE International Conference on Computer Vision*, pp. 370–378, 2015.
- [12] C. Ledig, L. Theis, F. Huszár, J. Caballero, A. Cunningham, A. Acosta, A. Aitken, A. Tejani, J. Totz, Z. Wang, *et al.*, “Photo-realistic single image super-resolution using a generative adversarial network,” *arXiv preprint*, 2017.

- [13] B. Lim, S. Son, H. Kim, S. Nah, and K. Mu Lee, “Enhanced deep residual networks for single image super-resolution,” in *Proceedings of the IEEE conference on computer vision and pattern recognition workshops*, pp. 136–144, 2017.
- [14] S. C. Park, M. K. Park, and M. G. Kang, “Super-resolution image reconstruction: a technical overview,” *IEEE signal processing magazine*, vol. 20, no. 3, pp. 21–36, 2003.
- [15] S. Rhee and M. G. Kang, “Discrete cosine transform based regularized high-resolution image reconstruction algorithm,” *Optical Engineering*, vol. 38, no. 8, pp. 1348–1356, 1999.
- [16] X. Marichal, W.-Y. Ma, and H. Zhang, “Blur determination in the compressed domain using dct information,” in *Proceedings 1999 International Conference on Image Processing (Cat. 99CH36348)*, vol. 2, pp. 386–390, IEEE, 1999.
- [17] H. Demirel and G. Anbarjafari, “Image resolution enhancement by using discrete and stationary wavelet decomposition,” *IEEE transactions on image processing*, vol. 20, no. 5, pp. 1458–1460, 2010.
- [18] T. Guo, H. Seyed Mousavi, T. Huu Vu, and V. Monga, “Deep wavelet prediction for image super-resolution,” in *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, July 2017.
- [19] H. Huang, R. He, Z. Sun, and T. Tan, “Wavelet-srnet: A wavelet-based cnn for multi-scale face super resolution,” in *The IEEE International Conference on Computer Vision (ICCV)*, Oct 2017.
- [20] S. P. Nanavati and P. K. Panigrahi, “Wavelets: applications to image compression-i,” *Resonance*, vol. 10, no. 2, pp. 52–61, 2005.
- [21] M. K. Mandal, S. Panchanathan, and T. Aboulnasr, “Wavelets for image compression,” in *Proceedings of IEEE-SP International Symposium on Time-Frequency and Time-Scale Analysis*, pp. 338–341, IEEE, 1994.
- [22] S. Grgic, K. Kers, and M. Grgic, “Image compression using wavelets,” in *ISIE’99. Proceedings of the IEEE International Symposium on Industrial Electronics (Cat. No. 99TH8465)*, vol. 1, pp. 99–104, IEEE, 1999.
- [23] S. Grgic, M. Grgic, and B. Zovko-Cihlar, “Performance analysis of image compression using wavelets,” *IEEE Transactions on industrial electronics*, vol. 48, no. 3, pp. 682–695, 2001.
- [24] R. Besar, C. Eswaran, S. Sahib, and R. Simpson, “On the choice of the wavelets for ecg data compression,” in *2000 IEEE International Conference on Acoustics, Speech, and Signal Processing. Proceedings (Cat. No. 00CH37100)*, vol. 6, pp. 3614–3617, IEEE, 2000.

- [25] V. Bairagi and A. Sapkal, “Selection of wavelets for medical image compression,” in *2009 International Conference on Advances in Computing, Control, and Telecommunication Technologies*, pp. 678–680, IEEE, 2009.
- [26] S. Banerjee, T. Hemraj-Benny, and S. S. Wong, “Covalent surface chemistry of single-walled carbon nanotubes,” *Advanced Materials*, vol. 17, no. 1, pp. 17–29, 2005.
- [27] R. S. Ruoff, D. Qian, and W. K. Liu, “Mechanical properties of carbon nanotubes: theoretical predictions and experimental measurements,” *Comptes Rendus Physique*, vol. 4, no. 9, pp. 993–1008, 2003.
- [28] J. Chen, A. M. Rao, S. Lyuksyutov, M. E. Itkis, M. A. Hamon, H. Hu, R. W. Cohn, P. C. Eklund, D. T. Colbert, R. E. Smalley, *et al.*, “Dissolution of full-length single-walled carbon nanotubes,” *The Journal of Physical Chemistry B*, vol. 105, no. 13, pp. 2525–2528, 2001.
- [29] K. B. Shelimov, R. O. Esenaliev, A. G. Rinzler, C. B. Huffman, and R. E. Smalley, “Purification of single-wall carbon nanotubes by ultrasonically assisted filtration,” *Chemical physics letters*, vol. 282, no. 5-6, pp. 429–434, 1998.
- [30] R. Baraniuk, M. Davenport, R. DeVore, and M. Wakin, “A simple proof of the restricted isometry property for random matrices,” *Constructive Approximation*, vol. 28, no. 3, pp. 253–263, 2008.
- [31] S. Mallat and Z. Zhang, “Matching pursuit with time-frequency dictionaries,” tech. rep., Courant Institute of Mathematical Sciences New York United States, 1993.
- [32] J. A. Tropp and A. C. Gilbert, “Signal recovery from random measurements via orthogonal matching pursuit,” *IEEE Transactions on information theory*, vol. 53, no. 12, pp. 4655–4666, 2007.
- [33] T. Blumensath and M. E. Davies, “Iterative hard thresholding for compressed sensing,” *Applied and computational harmonic analysis*, vol. 27, no. 3, pp. 265–274, 2009.
- [34] Y. Xiao, J. Yang, and X. Yuan, “Alternating algorithms for total variation image reconstruction from random projections,” *Inverse Problems and Imaging*, vol. 6, no. 3, pp. 547–563, 2012.
- [35] J. Mairal, F. Bach, J. Ponce, G. Sapiro, and A. Zisserman, “Non-local sparse models for image restoration,” in *Computer Vision, 2009 IEEE 12th International Conference on*, pp. 2272–2279, IEEE, 2009.
- [36] R. G. Baraniuk, V. Cevher, M. F. Duarte, and C. Hegde, “Model-based compressive sensing,” *IEEE Transactions on Information Theory*, vol. 56, no. 4, pp. 1982–2001, 2010.
- [37] D. Sundararajan, *The discrete Fourier transform: theory, algorithms and applications*. World Scientific, 2001.

- [38] S. Winograd, “On computing the discrete fourier transform,” *Mathematics of computation*, vol. 32, no. 141, pp. 175–199, 1978.
- [39] N. Ahmed, T. Natarajan, and K. R. Rao, “Discrete cosine transform,” *IEEE transactions on Computers*, vol. 100, no. 1, pp. 90–93, 1974.
- [40] W. Philips, “The lossless dct for combined lossy/lossless image coding,” in *Proceedings 1998 International Conference on Image Processing. ICIP98 (Cat. No. 98CB36269)*, pp. 871–875, IEEE, 1998.
- [41] M. Antonini, M. Barlaud, P. Mathieu, and I. Daubechies, “Image coding using wavelet transform,” *IEEE Transactions on image processing*, vol. 1, no. 2, pp. 205–220, 1992.
- [42] S. Sridhar, P. R. Kumar, and K. Ramanaiah, “Wavelet transform techniques for image compression-an evaluation,” *International journal of image, graphics and signal processing*, vol. 6, no. 2, p. 54, 2014.
- [43] T. F. Chan and H.-M. Zhou, “Total variation improved wavelet thresholding in image compression,” in *Proceedings 2000 International Conference on Image Processing (Cat. No. 00CH37101)*, vol. 2, pp. 391–394, IEEE, 2000.
- [44] A. Krizhevsky, I. Sutskever, and G. E. Hinton, “Imagenet classification with deep convolutional neural networks,” in *Advances in neural information processing systems*, pp. 1097–1105, 2012.
- [45] W. Ouyang, X. Wang, X. Zeng, S. Qiu, P. Luo, Y. Tian, H. Li, S. Yang, Z. Wang, C.-C. Loy, *et al.*, “Deepid-net: Deformable deep convolutional neural networks for object detection,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 2403–2412, 2015.
- [46] Y. Sun, Y. Chen, X. Wang, and X. Tang, “Deep learning face representation by joint identification-verification,” in *Advances in neural information processing systems*, pp. 1988–1996, 2014.
- [47] D. Liu, Z. Wang, B. Wen, J. Yang, W. Han, and T. S. Huang, “Robust single image super-resolution via deep networks with sparse prior,” *IEEE Transactions on Image Processing*, vol. 25, no. 7, pp. 3194–3207, 2016.
- [48] K. Kulkarni, S. Lohit, P. Turaga, R. Kerviche, and A. Ashok, “Reconnet: Non-iterative reconstruction of images from compressively sensed measurements,” in *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 449–458, IEEE, 2016.
- [49] Y. Zhang, Y. Tian, Y. Kong, B. Zhong, and Y. Fu, “Residual dense network for image super-resolution,” in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 2472–2481, 2018.
- [50] E. J. Candès *et al.*, “Compressive sampling,” in *Proceedings of the international congress of mathematicians*, vol. 3, pp. 1433–1452, Madrid, Spain, 2006.

- [51] Y. Bengio, P. Simard, and P. Frasconi, “Learning long-term dependencies with gradient descent is difficult,” *IEEE transactions on neural networks*, vol. 5, no. 2, pp. 157–166, 1994.
- [52] K. He, X. Zhang, S. Ren, and J. Sun, “Deep residual learning for image recognition,” in *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2016.
- [53] J. Kim, J. Kwon Lee, and K. Mu Lee, “Deeply-recursive convolutional network for image super-resolution,” in *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2016.
- [54] Q. Huynh-Thu and M. Ghanbari, “The accuracy of psnr in predicting video quality for different video scenes and frame rates,” *Telecommunication Systems*, vol. 49, no. 1, pp. 35–48, 2012.
- [55] A. Mousavi, A. B. Patel, and R. G. Baraniuk, “A deep learning approach to structured signal recovery,” in *Communication, Control, and Computing (Allerton), 2015 53rd Annual Allerton Conference on*, pp. 1336–1343, IEEE, 2015.
- [56] K. He, X. Zhang, S. Ren, and J. Sun, “Deep residual learning for image recognition,” in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 770–778, 2016.
- [57] R. Timofte, E. Agustsson, L. Van Gool, M.-H. Yang, and L. Zhang, “Ntire 2017 challenge on single image super-resolution: Methods and results,” in *Proceedings of the IEEE conference on computer vision and pattern recognition workshops*, pp. 114–125, 2017.
- [58] Y. Zhang, K. Li, K. Li, L. Wang, B. Zhong, and Y. Fu, “Image super-resolution using very deep residual channel attention networks,” in *Proceedings of the European Conference on Computer Vision (ECCV)*, pp. 286–301, 2018.
- [59] W. Swastika, M. F. Ariyanto, H. Setiawan, and P. L. T. Irawan, “Appropriate cnn architecture and optimizer for vehicle type classification system on the toll road,” in *Journal of Physics: Conference Series*, vol. 1196, p. 012044, IOP Publishing, 2019.
- [60] E. D. Cubuk, B. Zoph, D. Mane, V. Vasudevan, and Q. V. Le, “Autoaugment: Learning augmentation policies from data,” *arXiv preprint arXiv:1805.09501*, 2018.
- [61] C. Shorten and T. M. Khoshgoftaar, “A survey on image data augmentation for deep learning,” *Journal of big data*, vol. 6, no. 1, pp. 1–48, 2019.
- [62] M. D. Bloice, P. M. Roth, and A. Holzinger, “Biomedical image augmentation using augmentor,” *Bioinformatics*, vol. 35, no. 21, pp. 4522–4524, 2019.
- [63] Y.-D. Zhang, Z. Dong, X. Chen, W. Jia, S. Du, K. Muhammad, and S.-H. Wang, “Image based fruit category classification by 13-layer deep convolutional neural network and data augmentation,” *Multimedia Tools and Applications*, vol. 78, no. 3, pp. 3613–3632, 2019.

- [64] M. Elgendi, M. U. Nasir, Q. Tang, D. Smith, J.-P. Grenier, C. Batte, B. Spieler, W. D. Leslie, C. Menon, R. R. Fletcher, *et al.*, “The effectiveness of image augmentation in deep learning networks for detecting covid-19: A geometric transformation perspective,” *Frontiers in Medicine*, vol. 8, p. 629134, 2021.