

© 2016 Benjamin L. Delay

MACHINE LEARNING TECHNIQUES FOR IDENTIFYING RAILROAD  
BALLAST DEGRADATION

BY

BENJAMIN L. DELAY

THESIS

Submitted in partial fulfillment of the requirements  
for the degree of Master of Science in Electrical and Computer Engineering  
in the Graduate College of the  
University of Illinois at Urbana-Champaign, 2016

Urbana, Illinois

Adviser:

Professor Narendra Ahuja

# ABSTRACT

Railroad ballast is a layer of uniform sized crushed aggregate particles placed between, below and around the crossties. Railroad ballast transfers the load from crossties to the subgrade layer, provides lateral track stability and facilitates the drainage of water. Repeated traffic loading and environmental factors cause particle breakage, abrasion and polishing, which eventually degrade the ballast and result in fouling conditions. Traditional ballast fouling assessment includes manual sampling and identifying particle size distributions using sieve analysis. Recently, automatic ballast sampling (ABS) methods have been introduced to the railroad industry to obtain a sample of ballast and underlying layers using an approximately 1 m (3.28 ft.) long heavy duty steel tube driven into the ballast layer to depths of up to 2 m (6.56 ft.). Currently, visual-manual classification methods are used by experts to identify fouling conditions and degradation trends in the collected ballast samples. This thesis presents multiple approaches developed for the objective classification of ballast degradation using a combination of advanced machine vision and machine learning techniques. Initially, various computer vision algorithms are used to generate features associated with images of ballast cross sections at different degradation levels. Next, the generated features are used alongside a visual classification database provided by experts to develop, train, validate, and test a feedforward artificial neural network (ANN) using a supervised learning method. This work is further extended by implementing convolutional neural networks (CNNs) to serve as automatic feature generators. Finally, this approach is used on another cross-sectional ballast dataset that more closely resembles the type of ballast cross sections that can be found in the field. The findings of this study show that the proposed CNNs with an optimized topology can successfully classify ballast fouling in an effective and repeatable fashion with reasonable error levels. Further improvement of this technology holds the potential to

provide a tool for consistent and automated ballast inspection and life cycle analysis intended to improve the safety and network reliability of US railroad transportation systems.

*To my parents, for their love and support.*

# ACKNOWLEDGMENTS

I would like to thank Professor Narendra Ahuja for overseeing all of this work as well as Dr. Maziar Moaveni, Postdoctoral Research Assistant, and John M. Hart, Principal Research Engineer, for their continual assistance on this research, which was conducted in the Computer Vision and Robotics Laboratory at the Beckman Institute. I also thank them for their dedication to the Transportation Research Board (TRB) Safety IDEA program and Association of American Railroads (AAR) and Transportation Technology Center Inc. (TTCI) Technology Outreach projects it encompassed. Additionally, I would like to thank all of the undergraduate assistants who made this work possible, a group including, but not limited to, Michael Qiu, Yifeng Chu, and Zixu Zhao. Thanks also go to Shengan Wang, a fellow graduate research assistant, for her work during the early stages of this project and her preprocessing of many of the images in this project.

# TABLE OF CONTENTS

LIST OF TABLES . . . . .	vii
LIST OF FIGURES . . . . .	viii
LIST OF ABBREVIATIONS . . . . .	x
CHAPTER 1 INTRODUCTION . . . . .	1
CHAPTER 2 DATA ACQUISITION AND PREPROCESSING . . . . .	4
2.1 Ballast Laser Surface Profiling . . . . .	4
2.2 Tube Ballast Imaging . . . . .	7
2.3 Ballast Cross Section Imaging . . . . .	11
CHAPTER 3 MACHINE LEARNING FOR BALLAST CLASSI- FICATION . . . . .	17
3.1 Ballast Laser Surface Profiling Machine Learning . . . . .	17
3.2 Tube Ballast Image Classification . . . . .	19
3.3 Cross Sectional Image Classification . . . . .	26
CHAPTER 4 RESULTS AND DISCUSSION . . . . .	28
4.1 Ballast Laser Surface Profiling Results . . . . .	28
4.2 Tube Ballast Image Results . . . . .	31
4.3 Cross Sectional Image Results . . . . .	34
CHAPTER 5 CONCLUSION . . . . .	39
REFERENCES . . . . .	41
APPENDIX A CROSS SECTIONAL BALLAST IMAGE RESULTS	43

# LIST OF TABLES

2.1	Guideline for Manual-Visual Ballast Classification . . . . .	11
4.1	Neural Network Confusion Matrix . . . . .	29
4.2	Gaussian Mixture Model Confusion Matrix . . . . .	29
4.3	Mixed Small Video Frame GMM Confusion Matrix . . . . .	30
4.4	Accuracy of ABS Ballast Tube Sample Evaluation Within Test Dataset Using Feature Generation and Classification . . .	32
4.5	Accuracy of ABS Ballast Tube Samples Within Test Dataset Using Transfer Learning . . . . .	33
4.6	Accuracy of ABS Ballast Tube Samples Within Test Dataset Using CNN Training and Testing . . . . .	34



# LIST OF FIGURES

2.1	Sample Ballast Collection in Tray . . . . .	5
2.2	Sample Laser Line Deformation . . . . .	6
2.3	Automatic Ballast Sampler . . . . .	8
2.4	Tube Ballast Classification and Preprocessing . . . . .	9
2.5	Full Collection Procedure . . . . .	13
2.6	Ballast Tie Particle Distribution Size Curve . . . . .	13
2.7	Trench Images . . . . .	14
2.8	Strip Image Creation Process . . . . .	16
3.1	Laser Classification ANN . . . . .	17
3.2	One Hot Encoding Example . . . . .	18
3.3	Example Fourier Transform Mappings . . . . .	22
3.4	AlexNet Architecture and Sample Filters . . . . .	25
3.5	Example of Leave-One-Out Cross Validation . . . . .	26
4.1	Minimization of Objective Function . . . . .	35
4.2	Example Cross Sectional Result . . . . .	36
4.3	Example Error Graphs . . . . .	37
A.1	1-1050-1 Particle Size Distribution Characteristics . . . . .	43
A.2	1-1050-1 Error Characteristics . . . . .	44
A.3	1-1050-2 Particle Size Distribution Characteristics . . . . .	44
A.4	1-1050-2 Error Characteristics . . . . .	45
A.5	2-1308-1 Particle Size Distribution Characteristics . . . . .	45
A.6	2-1308-1 Error Characteristics . . . . .	46
A.7	2-1308-2 Particle Size Distribution Characteristics . . . . .	46
A.8	2-1308-2 Error Characteristics . . . . .	47
A.9	3-1354-C1-noball Particle Size Distribution Characteristics . . . . .	47
A.10	3-1354-C1-noball Error Characteristics . . . . .	48
A.11	3-1354-C1-Wall2-Panoramic Particle Size Distribution Char- acteristics . . . . .	48
A.12	3-1354-C1-Wall2-Panoramic Error Characteristics . . . . .	49
A.13	3-1354-C2-Wall2-Panoramic Particle Size Distribution Char- acteristics . . . . .	49
A.14	3-1354-C2-Wall2-Panoramic Error Characteristics . . . . .	50

A.15 3-1354-I1-noball Particle Size Distribution Characteristics . . .	50
A.16 3-1354-I1-noball Error Characteristics . . . . .	51
A.17 3-1396-1 Particle Size Distribution Characteristics . . . . .	51
A.18 3-1396-1 Error Characteristics . . . . .	52
A.19 3-1396-2 Particle Size Distribution Characteristics . . . . .	52
A.20 3-1396-2 Error Characteristics . . . . .	53
A.21 4-1460-1 Particle Size Distribution Characteristics . . . . .	53
A.22 4-1460-1 Error Characteristics . . . . .	54
A.23 4-1460-2 Particle Size Distribution Characteristics . . . . .	54
A.24 4-1460-2 Error Characteristics . . . . .	55
A.25 5-1557-1 Particle Size Distribution Characteristics . . . . .	55
A.26 5-1557-1 Error Characteristics . . . . .	56
A.27 5-1557-2 Particle Size Distribution Characteristics . . . . .	56
A.28 5-1557-2 Error Characteristics . . . . .	57

# LIST OF ABBREVIATIONS

ANN	Artificial Neural Network
CNN	Convolutional Neural Network
GMM	Gaussian Mixture Model
RNN	Recurrent Neural Network
SVM	Support Vector Machine

# CHAPTER 1

## INTRODUCTION

Railway ballast fouling is a problem that has primarily been tackled in civil engineering literature. Typically defined as the contamination of the ballast layer as a result of particle breakage and abrasion, migration of fine-grained soil from the subgrade, or introduction of coal dust from overloaded freight cars, the study of ballast fouling is the analysis of how to quantify and decide whether or not ballast needs maintenance. This problem is significant because ballast serves as the structural foundation for the rail lines on which trains run. When the ballast becomes excessively fouled it can cause major safety concerns and interruptions to rail service. In fact, past derailments have been directly attributed to ballast fouling and the resulting loss of track strength and stability [1]. Fouling is inevitable over time, as regular train operation contributes significantly to fouling (76% of fouling is directly attributable to breakdown of the ballast due to stress [2]). For all of these reasons, it is important to have the ability to rapidly and reliably identify areas of fouled ballast.

Classically, rail ballast fouling has been quantified by a variety of different metrics. The two most common are the Selig Fouling Index and the Percentage Fouling [2]. The Selig Fouling Index is the summation of percentage by weight of ballast material passing through 4.75mm and 0.075mm sieves used to separate rail ballast. The Percentage Fouling is the ratio of dry weight material passing a 9.5mm sieve to the dry weight of the total sample. Other more subjective measures of ballast fouling include visual inspection and ballast sampling and testing. Previous work done by the Civil Engineering Department and Beckman Computer Vision group at the University of Illinois has focused on this sampling and testing by looking at individual pieces of ballast and assessing them for various features like angularity and size in order to determine their suitability as ballast [3]. That research spurred investigation into other forms of automated inspection of ballast, including

the classification techniques presented in this thesis.

Initially, the Beckman group explored techniques involving image segmentation and the calculation of an image based fouling index (IBFI) from that segmentation [4]. While the IBFI values calculated in that research corresponded relatively well to the actual Selig Fouling Index values, the approach itself suffered from a few key issues. The segmentation required the researcher to manipulate a variety of parameters in order to achieve a good-looking result, the segmentation encountered issues when a large amount of particularly fine ballast was present, and the segmentation took a fair amount of time. Due to these issues, automation of this process proved fairly difficult, as subjectively satisfactory segmentation (in a visual sense) was critical to achieving good results.

In contrast, the main approach taken by this research is to use machine vision and machine learning techniques to determine the level of fouling in railroad ballast without the need for lab testing and ballast removal. To that end, a variety of data sources have been collected and analyzed, each in the hopes that there would be a correlation between the current analysis of the ballast and the automated analysis. If such a correlation exists, it may allow for machine vision and learning techniques to serve as a fast, repeatable, consistent metric by which to evaluate railroad ballast. It would make sense for this connection to exist, given that many of the phenomena that show ballast degradation are noticeable visually (factors like rock sharpness, texture, size, and the number of discrete pieces of ballast).

The research itself is broken into three main sections, each with three subsections based on the particular data source being examined. The three main sections are:

- Data Acquisition and Preprocessing
- Machine Learning for Ballast Classification
- Results of Ballast Classification

Each section covers the relevant topic for three different data sources. The data sources are:

- Ballast Laser Surface Profiling

- Tube Ballast Images
- Ballast Cross Section Images

The initial laser work classification was performed at the behest of Dr. Narendra Ahuja for a class, and its results led to the acquisition of other data sources. The Tube Ballast Images and Cross-Sectional Images were both attempts to acquire more natural and more accurate targets for the machine learning algorithms used. Later data sources had more relevant ground truth information for the degree of ballast fouling. The eventual hope for this work was to create a classifier capable of accurately distinguishing between different levels of fouling with a small degree of deviance from both objective ground truth and subjective human visual evaluation.

# CHAPTER 2

## DATA ACQUISITION AND PREPROCESSING

The data in the experiments performed during this master's thesis came from three main sources. In the first experiment, the initial laser work, the ballast being scanned was from samples that had previously been sieved to ensure no pieces of ballast over or under certain sizes were present. This had been done with sieves of various sizes, three of which were used. The second experiment, that of the tube ballast imaging, used images provided by Dr. Phil Sharpe of the AECOM engineering firm. These images were of cylindrical tube bored into the rail ballast and then split in half. This tube typically had a depth of around 2 meters and provided a depth sample of the ballast at the location it was taken. The third and final experiment, the ballast cross sections, were images of a horizontal cross section rail ballast taken from a trench dug underneath the rail ties. These trenches were perpendicular to the ties and extended beneath them to a depth of roughly 4 or 5 feet.

### 2.1 Ballast Laser Surface Profiling

Work on this suite of ballast degradation projects began with an experiment performed in conjunction with ECE 544: Pattern Recognition. The goal of the final project was to use the machine learning knowledge accrued during the course to tackle a practical problem. Dr. Ahuja suggested that buckets of differently-sized ballast acquired by the civil engineering department would make for a good dataset. The Federal Railroad Administration had an interest in automating remote inspection of ballast for replacement monitoring, and previous experiments had been run on this ballast to partition it into buckets containing differently sized pieces.

Three different collections of ballast were used for this experiment.

- Ballast passing through a 1.50 inch sieve, but retained on a 1.00 inch

sieve

- Ballast passing through a 1.00 inch sieve, but retained on a 0.75 inch sieve
- Ballast passing through a 0.75 inch sieve, but retained on a 0.50 inch sieve



Figure 2.1: Sample Ballast Collection in Tray

This experiment differed from the previous work performed by members of the Civil Engineering Department at the University of Illinois in that



instead of sampling individual particles of ballast [3], ballast from one of the collections was laid out on a cart tray (see figure 2.1). A line scanning laser was then shone from above at a 45 degree angle onto the ballast and the tray cart was moved perpendicular to the laser. This caused the laser line to deform (note that these videos were taken in the dark, a sample deformation line can be seen in figure 2.2). The video camera taking the images had a resolution of 1920x1080.

The resulting images were thresholded to extract the positions of all the red pixels in the image. The maximum positions of the laser line pixel along each row were taken as the input features to an artificial neural network. These positions were taken for every frame in multiple videos. Three different sizes of ballast were used and the network was trained across six videos (2 of each size containing roughly 200 frames each) and tested on another to see if it could correctly identify which size group individual frames belonged to. In addition, the experiment examined whether a plurality of frames in each video were classified properly.

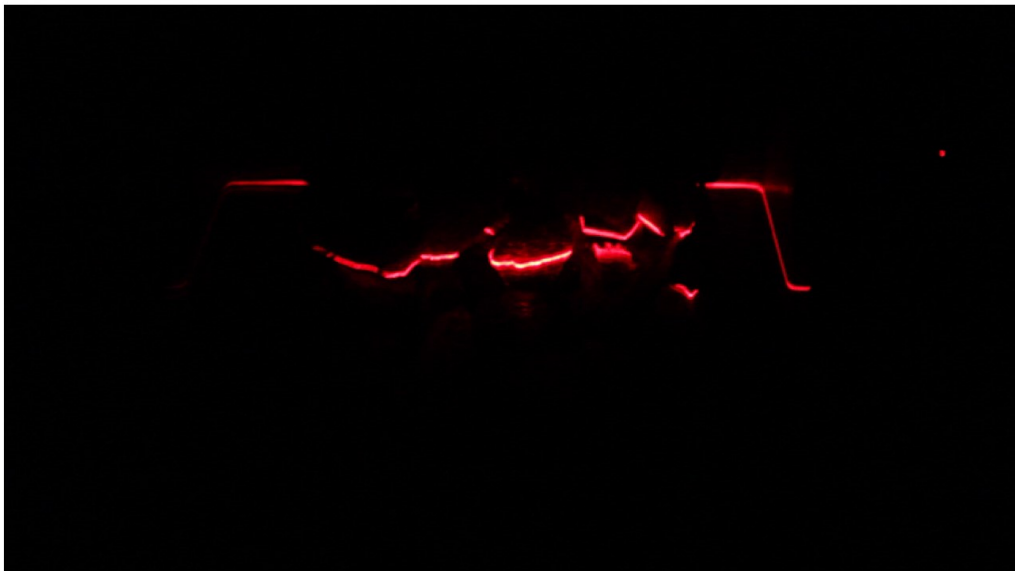


Figure 2.2: Sample Laser Line Deformation

The other features tested in this experiment include a discrete approximation of the first derivative of the laser positions (using a simple position difference vector). The use of this feature was an exploration of whether or

not the slope of the line might be more revealing of the ballast size than the position of the line.

## 2.2 Tube Ballast Imaging

The tube ballast images in this experiment were acquired using a relatively new technique of automatic ballast sampling pioneered by several different companies and researchers to facilitate the process of ballast sampling. This technique was first explored in a 2010 paper from Scott Wilson Pavement Engineering [5] that also included a proposed classification system for the resulting images. Tubes were driven into ballast using a hydraulic powered pneumatic hammer. The tubes were then extracted using a hydraulic jacking system. The tool used to do this is known as an automatic ballast sampler and can be seen in figure 2.3. The tubes themselves contain plastic liners which hold the samples. The liners are then extracted from the tube, split, and stored on a sample rack for later imaging. Each sample was then imaged with a 12 MP Nikon digital camera. These images were each captured with a constant 300 dots per inch spatial resolution.

These samples and images were initially used by Dr. Sharpe's company, AECOM, to provide assessments of ballast quality and analyze the type of ballast present in each of the samples. However, this work was done manually and required touching the ballast and having an expert note down a classification of each section of the tube. Motivated by a desire to simplify this workload, Dr. Sharpe provided a large database of these tube samples to our research group in the hopes that we could develop a method to automate some of this decision making.

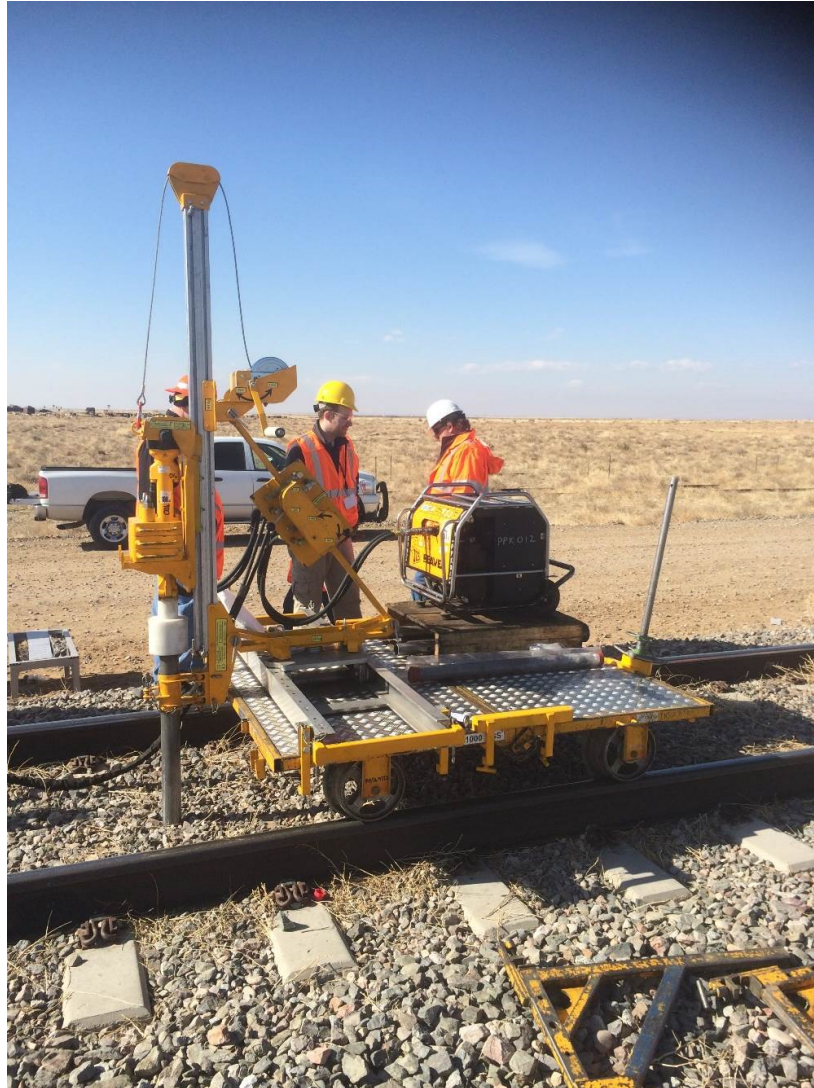


Figure 2.3: Automatic Ballast Sampler

Initial work began by determining that a simplification of the problem to ballast degradation detection (instead of the more general information generated by Phil Sharpe) was appropriate and still useful. The data samples from each database image were reclassified into five levels of degradation. A basic overview of the process can be seen in figure 2.4 and a more detailed textual description will follow.

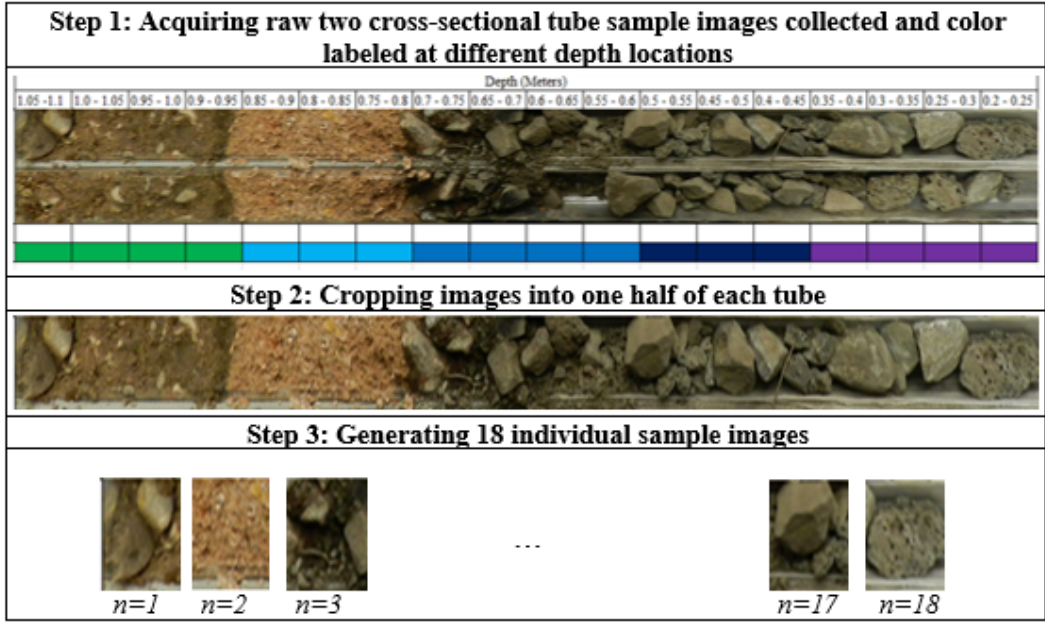


Figure 2.4: Tube Ballast Classification and Preprocessing

The initial images captured were 84 separate cross-sectional cropped images of ballast tube samples cut in half. Those images were further subdivided into 18 individual sections each, for a total of 3024 images. The choice of subdivisions was based on the quantification of class boundaries in the training data set since each image had 18 separate sections that were each classified.

These small segment images were used as the training data for the various methods used in this study and the labels were a number 1-5 corresponding with the segment class. The various methods proposed attempt to minimize the difference between the predicted label and the actual label across all the samples in the training set.

It should be noted that there were some issues with unequal numbers of training samples in the different classes in the tube ballast data set. This can become a problem when there are many more representatives of a certain class than there are of others. The network can easily get trapped in a local error minimum where the best policy is simply to label almost all input images as being of a certain class. If this occurs, the network may not learn any representation of the classes with a lower number of samples. Considering severe cases of ballast degradation are outliers (most ballast does not need replaced), this can be an issue.

A few different approaches were taken to deal with this issue when creating the training data:

- Upsampling - Images from the training set were taken and used to generate features. Once these features had been generated, a Gaussian distribution was fitted to the features of each label. New samples were generated by sampling at random from the Gaussian distribution using the theorem of inverse transformation [6]. This approach has issues when a Gaussian distribution does not accurately characterize the training data.
- Downsampling - A random subset of training images equal to the number of training images in the smallest class is selected from each class. Only these images are used to train the classifier. This is the easiest way of normalizing the size of the training classes, but can result in a very limited training set.
- Resampling - In this approach, features were randomly selected from smaller training classes to be included in the training set multiple times. This was done until the size of each class was equal to the size of the largest class. This avoids the problem of artificial sample generation and small training sets, but can introduce overfitting to specific samples within some of the smaller classes.

Note that these names should not be confused with the classic signal processing terms, but were instead techniques used to develop new features for use in training the network and hopefully eliminating the class skew. Table 2.1 gives more specifics about the distribution of class labels across all the images and what the assigned colors in some of the images refer to.

Table 2.1: Guideline for Manual-Visual Ballast Classification

<b>Ballast Condition</b>				
Clean	Slightly Dirty	Dirty	Very Dirty (Non-Cohesive)	Very Dirty (Slurried)
<b>Assigned Color</b>				
Magenta	Dark Blue	Medium Light	Light Blue	Dark Green
<b>Assigned Number</b>				
1	2	3	4	5
<b>Number of Labels</b>				
624	486	1210	574	80

### 2.3 Ballast Cross Section Imaging

Due to issues with the artificiality of the lab samples used in previous sections, it seemed prudent to try and apply similar machine learning techniques to images that more closely resemble those found in the field. In addition, it was necessary to acquire ground truth data for use in the classification of these images. The ground truth data in this case is a particle size distribution of the ballast samples, in which measurements are taken using 14 different sieves and the percentage of the sample (by weight) passing through each sieve is measured. From this data, the typical metrics used to measure degradation, Selig Fouling Index and Percentage Fouling, can be calculated. As a reminder, fouling index is the sum of the percentage by weight of ballast passing the 4.75mm sieve and the 0.075mm sieve. Percentage fouling is the ratio of the dry weight of material passing the 9.5mm sieve to the dry weight of the total sample. Because each of these metrics can be directly calculated from the more comprehensive particle size distribution, it was unnecessary to try and learn them directly, and instead the overall distribution was targeted for prediction.

The initial data was acquired in section 3 of the High Tonnage Loop at the Transportation Technology Center in Pueblo, Colorado. Photos were taken in trenches perpendicular to and underneath the rails at that location. These

trenches were created by removing the railroad ties, and digging the trench out with a front-end loader. The images themselves were acquired by lowering a 15.1 megapixel DLSR camera, using a positioning system supported by rails, into the trenches and marking out 24" by 16" sections of ballast with chalk lines and imaging them. The ballast in this marked section was then scraped into a 5 gallon bucket and collected for later lab analysis of the particle size distribution. The particular camera used for this work was a EOS Rebel T1i (500D) with CMOS sensors.

There were a few difficulties present in the collection of this data. Getting proper exposure on the photographs was a challenge given that sunlight was not necessarily evenly distributed on the trench wall. To this end, a tarp was used to eliminate some of the sharper shadow edge lines on the marked areas. Additionally, photos were taken at three different levels of exposure: one auto-calibrated by the camera, one using a longer exposure, and one using a shorter exposure. These precautions were taken because the areas could not be re-imaged after the trenches were again filled with ballast. It was also desired that the images have the same spatial resolution. In addition to the chalk size markings mentioned earlier, a white calibration ball 1" in diameter was used to verify that the images had a rough resolution of 80 pixels/cm.

The full collection process including sample images of a trench is detailed in figure 2.5. It should be noted that there were five of these trenches analyzed and in all fourteen 24" by 16" images were collected.

The exact sieving process was performed according to the ASTM C136 sieve analysis protocol [7]. The various distribution curves are presented in figure 2.6. The basic procedure behind this sieve analysis occurs in two stages. First, coarse fractions of the sample (coarse meaning particles with sizes above 12.7mm) are sieved using a sieve shaker. Then, the finer particles are sieved to determine the full distribution curve.

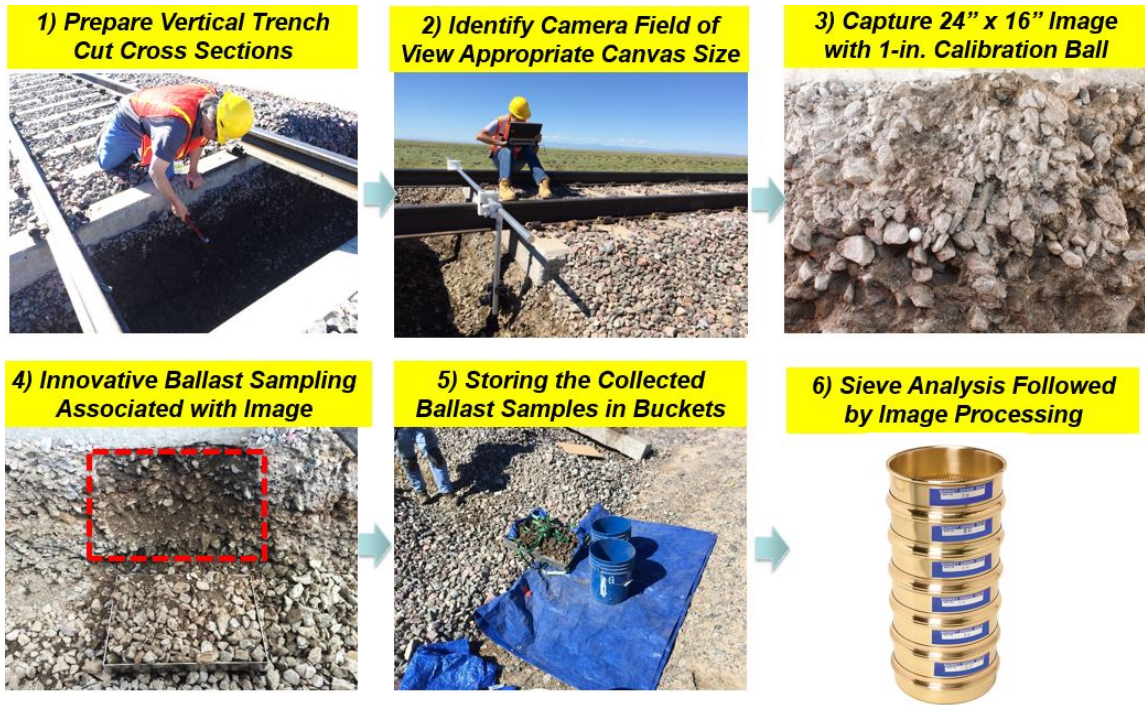


Figure 2.5: Full Collection Procedure

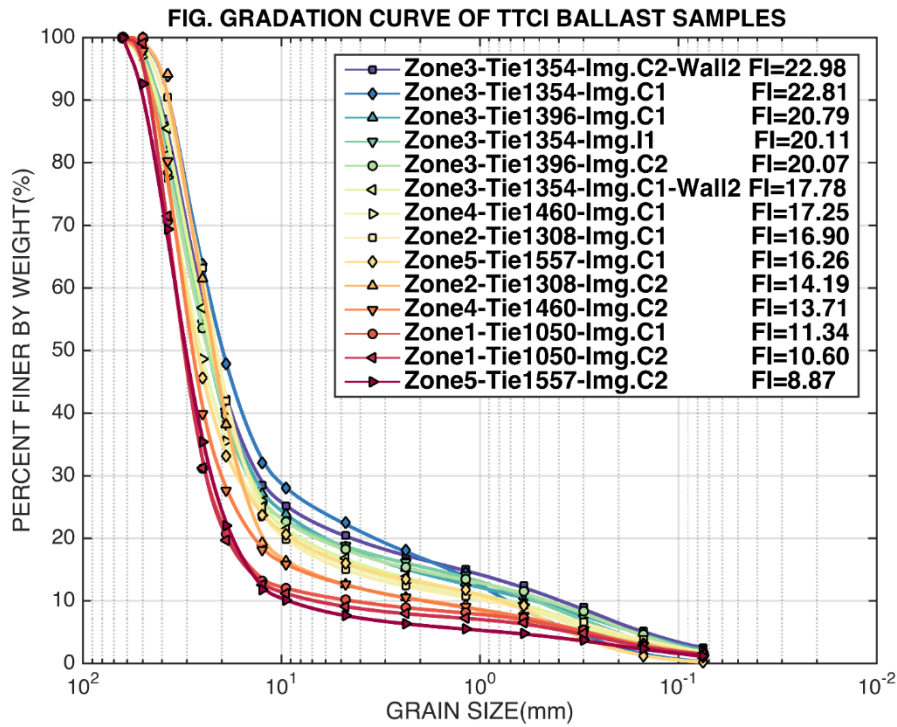


Figure 2.6: Ballast Tie Particle Distribution Size Curve



Images of the 14 trench image samples at the longest exposure level can be seen in figure 2.7. The associated Selig Fouling Index is also shown for each image and they are presented in order of increasing degradation level.















<b>Image ID = 5-1557-C2</b> FI = 8.87%	<b>Image ID = 1-1050-C2</b> FI = 10.6%	<b>Image ID = 1-1050-C1</b> FI = 11.34%
		
<b>Image ID = 4-1460-C2</b> FI = 13.71%	<b>Image ID = 2-1308-C2</b> FI = 14.19%	<b>Image ID = 5-1557-C1</b> FI = 16.26%
		
<b>Image ID = 2-1308-C1</b> FI = 16.9%	<b>Image ID = 4-1460-C1</b> FI = 17.25%	<b>Image ID = 3-1354-C1-Wall2</b> FI = 17.78%
		
<b>Image ID = 3-1396-C2</b> FI = 20.07%	<b>Image ID = 3-1354-I1</b> FI = 20.11%	<b>Image ID = 3-1396-C1</b> FI = 20.79%
		
<b>Image ID = 3-1354-C1</b> FI = 22.81%	<b>Image ID = 3-1354-C2-Wall2</b> FI = 22.98%	
		

Figure 2.7: Trench Images

Preprocessing of this dataset primarily consisted of figuring out how to generate a large number of data samples from the relatively small set of initial images. Because the degradation information can primarily be seen in a vertical ballast orientation (the ballast is typically more broken as the depth increases), image strips from each sample were taken and associated with the measured particle size distribution curves. The strip widths were determined by the expected input sizes of the convolutional neural networks used to classify the distribution (the architectures could have been changed to accommodate the new strip sizes, but it would have required retraining the networks to this new dataset and was not seen as necessary for the initial testing). These expected sizes were typically  $240 \times 240 \times 3$ , so each vertical strip across the higher resolution image was cut with horizontal lines every 240 pixels and these chunks were placed next to one another side by side. Figure 2.8 should clarify this entire process. Each initial image resulted in roughly 300 modified strip images (exact figures vary due to slightly different input sizes for some of the neural networks used). These modified strip images were directly used as the training and testing data for each neural network.

Given that the vertical dimension of the target images for input to the neural network were 240 pixels high, it was known that at least 14 strips were needed to capture the full 3178 pixel height of a single strip from the large image. Knowing this allowed the width of a single vertical strip to be calculated simply by finding the largest multiple of 14 that was less than 240 (the width of the target image). In most cases, that meant each vertical strip was 17 pixels wide. Once all 14 strips had been placed in the target image (filling 238 of the 240 available columns in the image), the remaining  $240 \times 2$  area was filled with black pixels.

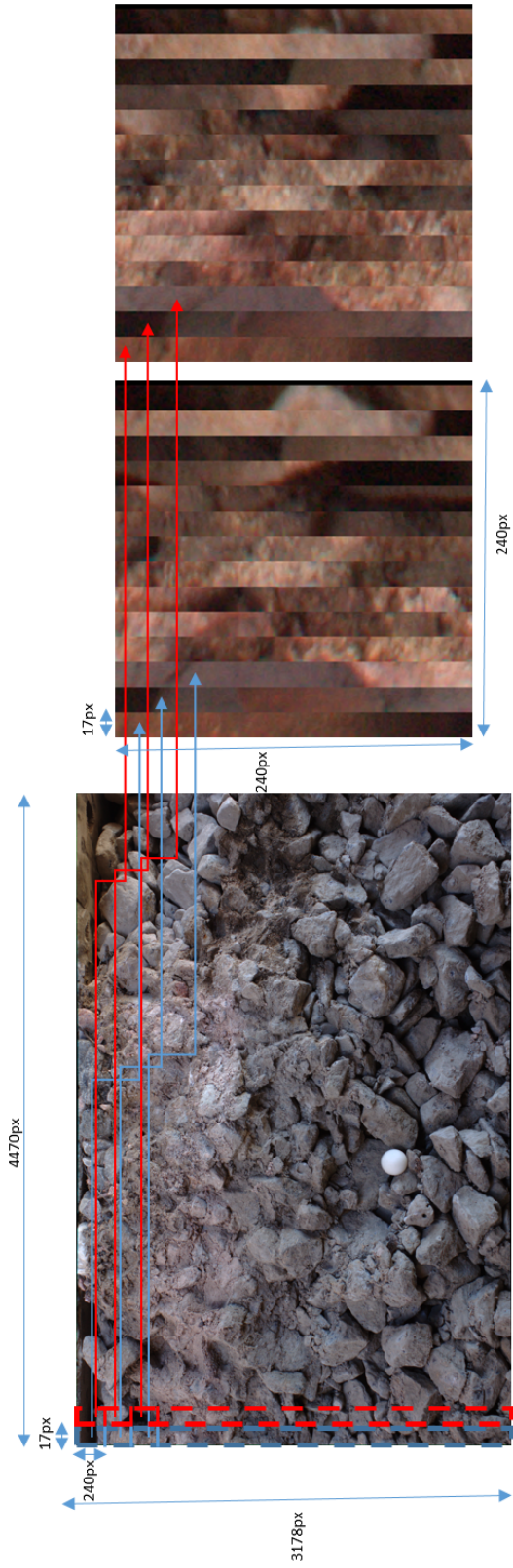


Figure 2.8: Strip Image Creation Process

# CHAPTER 3

## MACHINE LEARNING FOR BALLAST CLASSIFICATION

Like Chapter 2, this chapter is broken down into three main sections dealing with each of the separate datasets explored in this research. The approaches to each dataset were markedly different due to their unique labeling and constraints. It should be noted that large portions of the section on Tube Ballast Image classification have been submitted to and will be presented at the 2017 ASCE Geotechnical Frontiers conference along with a paper (see [8]). That content has been expanded upon here.

### 3.1 Ballast Laser Surface Profiling Machine Learning

The initial machine learning work for the classification of laser imagery was quite basic. The focus of the pattern recognition final project was more algorithm-oriented than result-oriented in that it focused on implementation and understanding more than success at the chosen task. As such, a basic artificial neural network was used (see figure 3.1).

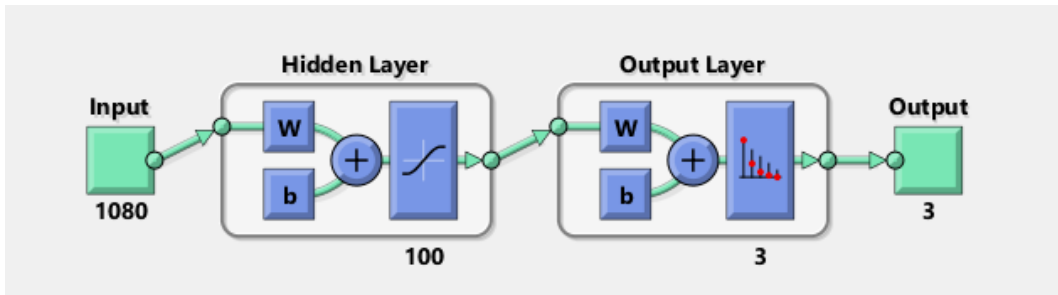


Figure 3.1: Laser Classification ANN

The 1080 represents the dimension of the input feature vectors being used (in this case each dimension was the maximum x position of the laser line at each y location in the image), the 100 represents the number of hidden

nodes, and the 3 represents the number of output classes. The w and b labels represent the weights and biases being trained in each node. The classifier attempted to minimize cross entropy loss (as defined in [9]) of the form

$$C = -\frac{1}{n} \sum_x \sum_j [y_j \ln(a_j)] \quad (3.1)$$

where n is the number of samples, x represents summation over each sample, j represents summation over each class (where classes are in a one-hot representation [10], see figure 3.2),  $y_j$  is 1 if the sample is the correct target class, and a is the output of the neuron.

<b>Binary</b>	<b>One-hot</b>
<b>000</b>	<b>00000001</b>
<b>001</b>	<b>00000010</b>
<b>010</b>	<b>00000100</b>
<b>011</b>	<b>00001000</b>
<b>100</b>	<b>00010000</b>
<b>101</b>	<b>00100000</b>
<b>110</b>	<b>01000000</b>
<b>111</b>	<b>10000000</b>

Figure 3.2: One Hot Encoding Example

The actual focus of the pattern recognition paper, an analysis of stochastic gradient descent, is not particularly relevant to this thesis beyond being the method of gradient descent used by the back-propagation algorithm that minimizes the cross entropy loss across epochs. Back-propagation is how the network is actually trained and can be thought of as repeated application of the chain rule and gradient descent to update weights in the network to minimize the output error.

It should be noted that a Gaussian mixture model (GMM) Classifier was used to validate the performance of the neural network. Gaussian mixture models are “a parametric probability density function represented as a weighted sum of Gaussian component densities” [11]. This classifier was chosen because the coursework had used it previously and because it could provide a baseline for the artificial neural network performance. A comparison of the results of the GMM and ANN will be presented in a later chapter.

## 3.2 Tube Ballast Image Classification

The approaches taken to classifying the individual examples of tube ballast follow a fairly natural exploratory progression. Initially, it was thought that using some rather well known computer vision algorithms, it was ideal to manually generate features representative of the various classes of our data set (this is the method referred to below as Method 1). Once generated, these features were classified using an artificial neural network (ANN) configuration. Error for this classification (and in later methods) was defined as the percentage of samples labeled differently than the supervised label. Each classifier attempted to minimize this error. This method requires prior knowledge about what kind of features are representative and it also typically requires a large amount of parameter tweaking to generate the various feature sets. Both of these issues can make this sort of classification problematic. Unfortunately, it also produced results that were not very accurate (40% correct classification on average, with a few methods peaking around 45%), which spurred the move to different approaches that instead used convolutional neural networks (CNN). The ballast classification approaches using CNNs can be broken down into two distinct categories. The first (later referred to as Method 2) was to use CNNs pre-trained on the Imagenet dataset (a dataset of 1.4 million images of 1000 different object classes commonly used to test CNNs efficacy) to generate a probability distribution vector. This probability distribution vector was then used as a feature input to ANNs and support vector machines (the SVMs are there to serve as a means of validating the ANN accuracy) in hopes that it would be more representative of the images underlying class than the manually generated features. Ultimately though, this approach was simply a quick test to assess the differences between various CNN architectures and to give a baseline for the third approach, an approach also involving CNNs. The second CNN-based approach to our ballast classification problem (Method 3) was to train two of the CNN architectures mentioned previously on the raw images in our dataset. This method is significantly more time-consuming than the previous approaches. As such, only a couple of CNN architectures were chosen based on a combination of their size and their performance in the previous approach. It should be noted that this kind of training requires a lot of computing power, typically in the form of multiple high-end GPUs.

### 3.2.1 Method 1: Feature Generation and Classification

This first method involved generating features based on prior knowledge of what sorts of visual markers could be used to classify the dataset. These features were then fed into an artificial neural network classifying samples into five distinct classes. ANNs consist of a large number of neurons connected to one another, with each neuron having a set of associated weights (each weight can be thought of as the strength of a connection between that neuron and another neuron). These neurons are typically arranged in layers. The first layer is used as the input (features) and the last layer provides the output (or classification). Typically every neuron in a single layer is connected to each neuron of the next layer. The final layer typically consists of a number of neurons equal to the number of classes that need to be distinguished. A softmax function is generally applied to the final layer. This softmax function attempts to force the output of all but one neuron to zero, and one special neuron to 1. The node with a value of one is the predicted class of the sample. Essentially, the softmax layer is trying to reproduce the one-hot encodings used in defining the classes.

Once a network is set up it can be trained by deciding upon an appropriate error function and using an operation known as back-propagation. The cross entropy error criterion was utilized to improve performance during training (see equation 3.1).

It is important for this method that the features generated be the same size (in terms of dimensionality) across all the samples. Classification can be performed without that requirement, but it typically necessitates using some sort of dimensionality reduction technique on the dataset, which introduces additional possibilities for error. Therefore, the following features were chosen as good test features:

- Grayscale Histogram: This is simply a histogram of intensities in a grayscale image. The standard OpenCV grayscale conversion formula was used ( $Y = 0.299R + 0.587G + 0.114B$  where  $Y$  is the luminance, and  $R, G,$  and  $B$  are the pixel intensities from 0-255 of each color channel).
- Color Histogram: This is a histogram of intensities in a color image, and consists of individual histograms in each of the three color channels.

- **Fourier Spectrum Data:** First a 2D Fourier transform is applied to an image, and then the image is downsampled and unrolled (converted from a matrix to a vector by taking each element of the matrix column-wise) in order to reduce the dimensionality of the input feature.
- **Canny Edge Density:** A standard Canny edge detection formula is used on a grayscale image, and then the result is non-maximally suppressed. After that, a sum is taken of the number of pixels remaining along each row of the image (so essentially each edge pixel along a row). The summed value was then divided by the total number of pixels in the row. This resulted in a vector of the size of the height of the image, where each dimension of the vector was an edge density along the corresponding row.
- **Raw Grayscale Image:** A grayscale image was downsampled, unrolled, and then fed into the ANN as a feature.

The criteria for selecting the image features was primarily based on our knowledge of the human decision processes involved in visual classification of the ballast images. The features used by experts to generate the initial training sets were reviewed and roughly corresponding MV features were found.

A generic example of one of the features used that also illustrates the kind of information that feature captures can be seen in figure 3.3. This feature, much like the others, was chosen because of its intuitive visual qualities.

The top images in the figure are visual representations of rotated 2D sinusoidal functions (biased to be positive, as digital images have no negative values). The bottom images are the corresponding Fourier transforms. The dots represent the frequencies at which edges occur in various directions. In more complicated images, the transform does not result in simple dots, but a full black and white image where the pixel values represent the presence of edge frequencies across a large number of directions.



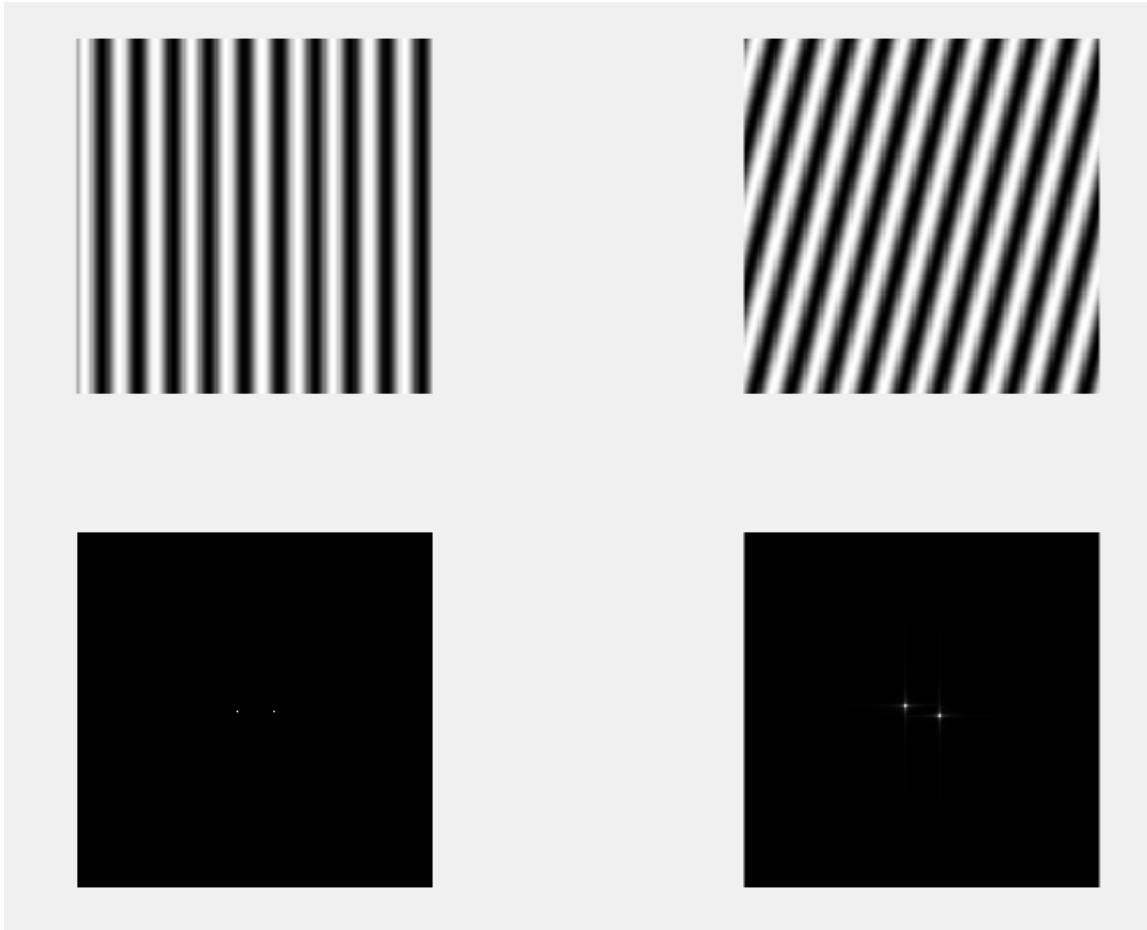


Figure 3.3: Example Fourier Transform Mappings

This image was not intended as a fully representative example of the image features used in Method 1, but as an intuitive example of the logic behind choosing features that map closely to visual phenomena. Most of the features chosen capture that information, though they do not lend themselves to nice visualization.

### 3.2.2 Method 2: Transfer Representation Learning

Method 2 was initially used as a quick method to test the viability of various artificial neural network configurations. It relied on using convolutional neural networks (CNNs) pre-trained on the ImageNet 2012 dataset [12] to generate the probability distribution vectors associated with that task. These probability vectors were 1000 dimensional vectors where each dimension represented the probability of the main subject of the image being of the class

of object represented by that dimension. The 1000 classes contained objects as disparate as dogs, trucks, velvet, and people. After being generated, these vectors were then used as features to ANNs and SVMs and used to classify the images into one of our five image classes [13]. The SVM classification accuracies were arrived at using 10 fold cross validation, which is a technique that splits datasets into training and validation sets 10 times and averages the accuracies on the validation sets across all 10 trials. This method was used in order to check the ANN performance, as it is more robust to easy testing set outliers.

CNNs are a type of feed-forward neural network that focus on arranging neurons so that they respond to overlapping regions of a signal in the same way that a human visual cortex might. They are called convolutional neural networks because the operation they perform is equivalent to the idea of sliding a window across an image (the convolution operation) and using the resulting tiled image as features to higher layers. This operation is able to update based on the resulting classification error of the network in much the same way that a more standard ANN updates. The motivations behind this second method were threefold:

- It allows a quick assessment of the viability and speed of different neural network configurations.
- The features generated in a CNN to distinguish between various image classes should have at least some crossover between tasks (detectors of low level features like corners, edges, squares, etc., are generically useful in image recognition and not necessarily dataset specific), and the final probability distribution vector might reflect the efficacy of these features [14].
- This method is much quicker than training a full CNN, a task which can take days on some of the deeper architectures, even with reasonably high-end hardware.

The first of these motivations is rather straightforward. The overall approach that methods 2 and 3 taken together constitute is known as fine-tuning. It typically involves taking a neural network that has been pre-trained on a task similar to the one currently being performed and using said networks weights and architecture to serve as a starting point for the

new problem [14]. However, in order to fine-tune an initial starting network has to be selected. Method 2 allows for quick assessment of these different networks in hopes of narrowing down the possible options.

The second motivation is usually captured under the label transfer representation learning [14]. It is the idea that similar features are useful for distinguishing disparate categories of images. Therefore, information that is used to classify large, but semantically unrelated datasets can also be useful for other classification tasks. To some degree, this approach also helps mitigate the size of our training set and the class size differences.

The third motivation is another practical one. This method can be performed for a single architecture in under a minute, while training a full CNN can take many hours (or days for deeper networks). This allows for a quicker turnaround time and allows for more architectures and approaches to be tested.

### 3.2.3 Method 3: CNN Training and Classification

The third methodology applied to this ballast identification problem is that of training a deep convolutional neural network to generate image features useful for classification of ballast. This method is an extension of the second method (the architectures used were chosen based on their performance under the second methodology in the hope that it would translate to success in this third method). However, now the network will be generating its own internal features instead of using features that are useful to generic image recognition. These internal features will be better suited to classifying the training set and that success should (assuming a representative training set) translate over to the test set. The training featured in this approach uses simple softmax layer and back-propagation to minimize the number of mislabeled samples. Back-propagation in a CNN is very similar in how it trains the network to the ANN implementation in that it works primarily through the calculation of error gradients for each node in the network and then updates to minimize that error [15]. The initial weights are the same as those in the architecture used in the second method.

The main motivation behind this approach to ballast identification is that features no longer need to be chosen by a human user, but are instead gener-

ated by the algorithm in question owing to their usefulness in identification. This often results in distinguishing features that humans recognize (things like a nose structure for face recognition), but often generates useful distinguishing features that a human may not have thought to replicate. The other major benefit is that this approach is systematic and independent of any preconceived bias on the part of the user. Once properly trained, the system can output an independent evaluation of the ballast degradation that is consistent and replicable.

The specific architecture chosen for this approach was Alexnet (see figure 3.4 for a visual representation of the Alexnet architecture and an example of the kind of information each layer might learn while classifying a more typical image) due to memory limitations with the GPU used for training the test set. As Alexnet is a reasonably shallow network (8 layers), it can be stored in a smaller amount of memory than the other networks and also trained quite quickly. While Alexnet performed poorly in the ANN classification in method 2, the hope was that this method would improve the results relative to the other Alexnet results. This could then lend support to the idea of training a deeper, more robust network.

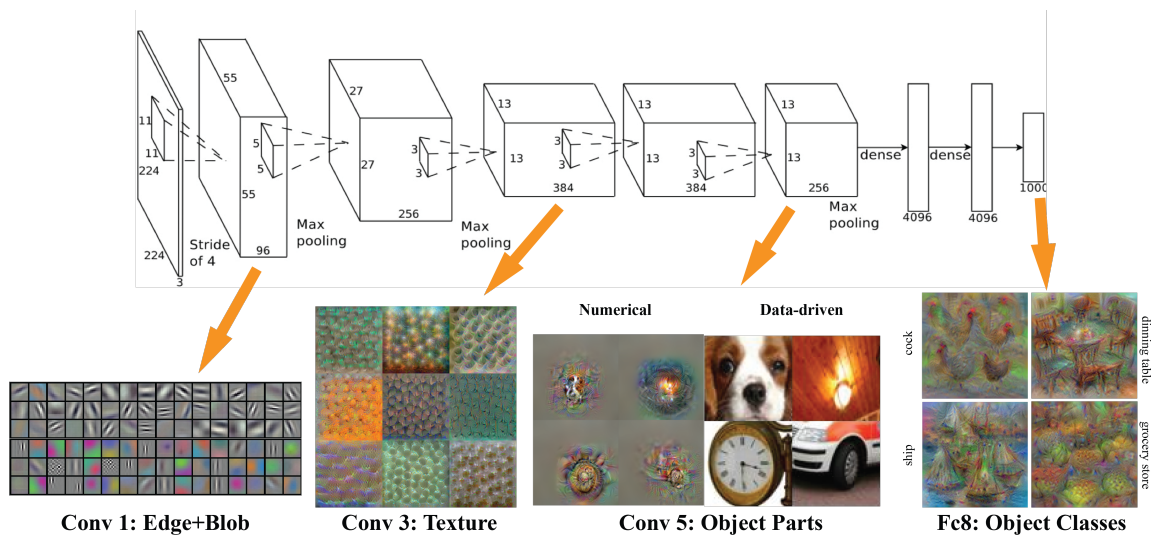


Figure 3.4: AlexNet Architecture and Sample Filters [16]

### 3.3 Cross Sectional Image Classification

The machine learning techniques applied to this new dataset closely resemble those applied to the tube ballast image set in methods 2 and 3. The novelty here is that this data set has actual ground truth data to target in the form of a particle size distribution graph associated with each image. Because the initial data consisted of only 14 different images, leave-one-out cross validation (defined in [17]) was also used to verify the results of the particle size distribution training without biasing the classifier.

Figure 3.5 shows an example of how leave-one-out cross validation works on a dataset. In this case, the dataset left out for testing corresponds to one image, while the datasets used are the other 13 images.

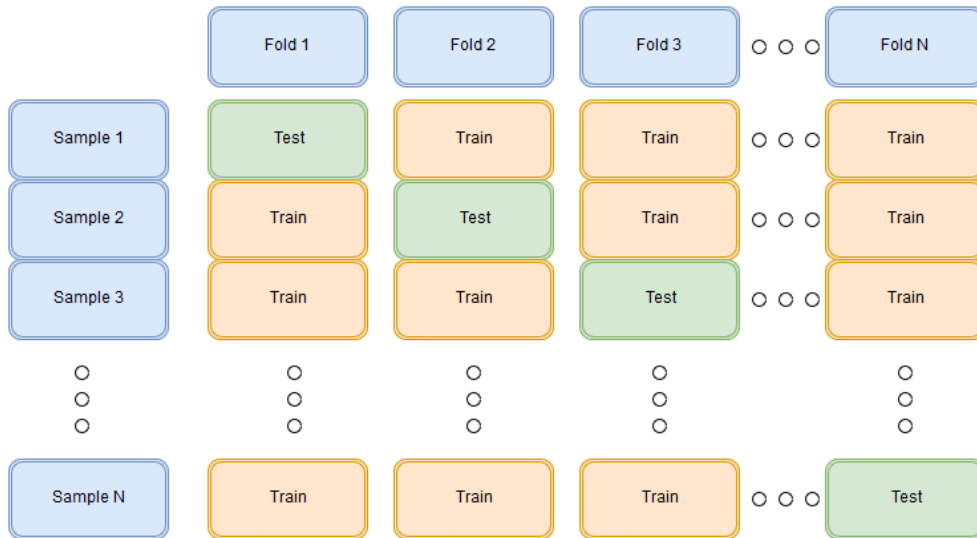


Figure 3.5: Example of Leave One Out Cross Validation

The method 2 equivalent for classifying the cross sectional images relied on using a CNN to generate 1000 dimensional probability vectors for all of the strip images relying on the justifications mentioned previously. Then, all of the probability distribution vectors associated with 13 of the initial images were used to train another classifier (in this case an artificial neural network) to predict the associated particle size distributions. Testing was then performed on the remaining set of probability distribution vectors associated with the 14th image. This was repeated 14 times, leaving a different image out of the training each time (so in total, there were 14 neural networks at the end). The training was performed with the goal of minimizing

the mean squared error across all the dimensions of all the 14 dimensional vectors being targeted. The test error was measured in a similar way, though for visual purposes the results were usually presented as a box chart distribution (to show where the mean guesses for given dimensions across all the strip images associated with a single initial image were). Mean percentage remaining guesses for each sieve size across all the strip images from each initial image were also used to display the results of the particle size distribution prediction.

# CHAPTER 4

## RESULTS AND DISCUSSION

As in the previous sections, the results and discussion section has been broken down and presented by covering each of the different data sources (and machine learning methods) separately. Most of the result presentation centers around a discussion of how well the various methods and data achieved their desired error metrics and classification results.

### 4.1 Ballast Laser Surface Profiling Results

The results for this data source and method of classification are rather simplistic and do not involve any targeted error metrics outside correct classification. Because the machine learning algorithm was simply seeking to separate the laser line in each image frame into one of three different size groups, the results are a 3x3 confusion matrix identifying how many and what misclassifications and correct classifications occurred.

The classes in the resulting confusion matrices use these class labels:

- Class 1 - Ballast passing through a 1.50 inch sieve, but not through a 1.00 inch sieve
- Class 2 - Ballast passing through a 1.00 inch sieve, but not through a 0.75 inch sieve
- Class 3 - Ballast passing through a 0.75 inch sieve, but not through a 0.50 inch sieve

The class labels in the first row represent the predicted classes while the class labels in the first column represent the actual class labels. A perfect confusion matrix would have every number inside along a diagonal in the correct location (indicating the predicted class was always matched with the

actual class). The rate of correct guesses is defined as the number of labels in the correct diagonal divided by the sum of labels in the row. The percent of false positives for a class is defined as the number of labels in the correct diagonal divided by the sum of the labels in the column.

Two confusion matrices are presented, one from a neural network and one from a Gaussian mixture model that serves as a point of comparison.

The confusion matrix for the laser classification can be seen in table 4.1, while the confusion matrix for the Gaussian mixture model can be seen in table 4.2.

Table 4.1: Neural Network Confusion Matrix

<b>Classes</b>	<b>1</b>	<b>2</b>	<b>3</b>	<b>Percent Correct</b>
<b>1</b>	43	1	5	87.8%
<b>2</b>	20	67	31	58.6%
<b>3</b>	3	0	61	95.3%
<b>Percent False Positives</b>	34.8%	1.5%	37.1 %	$\frac{74\% \text{ right}}{26\% \text{ wrong}}$

Table 4.2: Gaussian Mixture Model Confusion Matrix

<b>Classes</b>	<b>1</b>	<b>2</b>	<b>3</b>	<b>Percent Correct</b>
<b>1</b>	66	0	0	100%
<b>2</b>	10	58	0	85.3%
<b>3</b>	78	19	0	0.0%
<b>Percent False Positives</b>	57.1%	24.6%	0.0 %	$\frac{53.7\% \text{ right}}{46.3\% \text{ wrong}}$

The other criterion to consider for each of these confusion matrices is whether or not the consensus size pick in each video is correct. Fortunately, the neural network outputs the correct response in all three cases. On the



other hand, the GMM failed to accurately classify any of the videos into the third category. This is likely due to a large artifact caused by the initial positioning of the trolley. As the other video did not contain this artifact, it seems the GMM was not able to accurately train on either individual video. The neural network was able to overcome this bias, but the issue itself stems from a lack of training data. Inspection of the covariance matrix of the lines in each video seemed to confirm the artifact as the issue. The small video with the artifact had low variance in the higher dimensions of the covariance matrix (indicating little change in the position of the laser line) while the corresponding variance in those same dimensions across all five other videos was higher. By the time this issue was discovered it was too late to easily acquire more video data, so an attempt was made to mix frames from the two small rock videos to create the training and test sets.

The attempt at mixing the frames from the two smaller videos improved the situation, but hurt classification accuracy on the class 2 samples (see table 4.3).

Table 4.3: Mixed Small Video Frame GMM Confusion Matrix

<b>Classes</b>	<b>1</b>	<b>2</b>	<b>3</b>	<b>Percent Correct</b>
<b>1</b>	66	0	0	100%
<b>2</b>	9	7	52	10.3%
<b>3</b>	33	0	64	66.0%
<b>Percent False Positives</b>	38.9%	0.0%	44.8 %	$\frac{59.3\% \text{ right}}{40.7\% \text{ wrong}}$

In general, while these results were not a direct indicator that levels of ballast degradation could be accurately assessed by machine learning algorithms, they did provide justification for later analysis. The classifiers seemed clearly capable of at least distinguishing ballast size, which correlates quite strongly with degradation levels. Therefore, it was decided that more and better training data with actual ground truth degradation information was needed.

## 4.2 Tube Ballast Image Results

The results in this section are broken down by the method used to classify this source of data. There are three such methods: Feature Generation and Classification, Transfer Learning, and CNN Training and Testing.

### 4.2.1 Feature Generation and Classification Results

The results for method 1 are summarized in Table 4.4. The results from method 1 were not particularly great, which is what spurred the moves to later methods. None of the methods of generating synthetic data or balancing the size of the classes seemed to improve the overall classification accuracy, though the up-sampling method was promising for a few of the features. Overall, the color histogram and Fourier spectrum features seemed to perform better than other features.

This makes some intuitive sense, as color communicates a large amount of information and may vary quite strongly across classes. As far as the Fourier spectrum features go, the horizontal and vertical frequencies in the image most likely reflect the prevalence of edges and large or small objects in the image. It also makes some intuitive sense that these would allow for more accurate classification.

Unfortunately, the overall results from this classification do not initially seem particularly promising, as the accuracy of distinguishing between the various ballast degradation classes is quite low. One confounding factor in this analysis is that the targets for this classification are not objective ground truth targets (they are instead subjective human evaluations of fouling in a given area). Given that the lines between the different fouling classes are somewhat nebulous, near misclassifications are likely, and the human graders (whose evaluations were used for the initial training data) may experience some inconsistency in their classification. Additionally, while the classifier looks at each individual image section without additional relative depth context, the human graders had access to that information, which may have altered their guesses.

Table 4.4: Accuracy of ABS Ballast Tube Sample Evaluation Within Test Dataset Using Feature Generation and Classification

<b>Image Feature Used</b>	<b>Unmodified Accuracy(%)</b>	<b>Up-sampled Accuracy (%)</b>	<b>Down-sampled Accuracy (%)</b>	<b>Re-sampled Accuracy (%)</b>	<b>Average (%)</b>
<i>Canny Edge Density</i>	39.23	36.74	23.56	38.38	<b>34.48</b>
<i>Gray Histogram</i>	35.86	40.70	28.28	33.22	<b>34.52</b>
<i>Color Histogram</i>	44.44	40.44	37.78	35.57	<b>39.56</b>
<i>Fourier Spectrum</i>	45.29	37.25	38.11	34.59	<b>38.81</b>
<i>Raw Grayscale Image</i>	37.37	44.13	35.08	32.89	<b>37.37</b>
<b>Average (%)</b>	<b>40.44</b>	<b>39.85</b>	<b>32.56</b>	<b>34.93</b>	

In order to test this, the same grader who initially classified these degradation levels was asked to reevaluate and classify fifty of the tube ballast image sections (chosen randomly) a few months after the initial classification. This grader was only able to achieve an exact reclassification accuracy of 36%, though their average class error (defined as the distance between the predicted class and the actual class across all the samples, i.e. a prediction of class 1 when the actual class was class 5 would be an error of 4) was quite low (only .70, where completely random classification would result in an average error of 1.44). The classifier compares quite favorably to this result and this result seemingly demonstrates at least some aspects of ballast fouling can be seen.

However, these results also demonstrate that a better source of data with

objective ground truth results is necessary to accurately train a classifier, particularly when extraneous information about the ballast (depth, location, etc.) is not directly known. These results and this need are what spurred the acquisition of the cross-sectional ballast image dataset.

## 4.2.2 Transfer Learning Results

The results for method 2 are summarized in Table 4.5. This method yielded quite an improvement over the first method in terms of overall classification accuracy. It also gives some idea of which networks would have the best overall performance if fully trained on the tube ballast dataset. There are some strange caveats to note about these results, such as the SVM classification accuracy being on average higher than the ANN accuracy, but this may simply be a result of poor training trials for the ANNs since the network size as well as topology could be further optimized using trial and error method. It could also simply be that the test set chosen for the networks which was kept consistent across all networks was difficult to classify.

Table 4.5: Accuracy of ABS Ballast Tube Samples Within Test Dataset Using Transfer Learning

<b>Network</b>	<b>SVM Cross Validated Accuracy(%)</b>	<b>ANN Test Accuracy (%)</b>	<b>Average Network Accuracy (%)</b>
<i>AlexNet</i>	49.2	43.7	<b>46.5</b>
<i>GoogleNet</i>	48.7	46.9	<b>47.8</b>
<i>Vgg16</i>	51.0	48.9	<b>50.0</b>
<i>Vgg19</i>	51.5	51.3	<b>51.4</b>
<i>VggF</i>	50.9	49.6	<b>50.3</b>
<i>VggM2048</i>	50.1	51.6	<b>50.9</b>
<i>VggM</i>	51.5	49.6	<b>50.6</b>
<i>VggS</i>	51.0	48.0	<b>49.5</b>
<b>Average Classifier Accuracy (%)</b>	<b>50.5</b>	<b>48.7</b>	

### 4.2.3 CNN Training and Testing Results

The results for method 3 are summarized in Table 4.6. The CNN training and testing method can also be thought of as an overall improvement, though it could be far extended past what is presented in this thesis.

The CNN training and testing method can also be thought of as an overall success, though it could be far extended past what has been done here. The network that was trained, Alexnet, showed a relative improvement of nearly 10% over its performance in the transfer learning stage (the validation error here can be thought of as the test error in the transfer learning stage). The training of this network only took roughly 20 minutes for 40 epochs, and the number of epochs could easily be extended. Figure 4.1 shows the minimization of the objective function on both the training and validation sets over the epochs.

Table 4.6: Accuracy of ABS Ballast Tube Samples Within Test Dataset Using CNN Training and Testing

<b>Network</b>	<b>End Training Accuracy(%)</b>	<b>End Validation Accuracy (%)</b>
<i>AlexNet</i>	55.1	51.0

## 4.3 Cross Sectional Image Results

The results on the cross sectional image strips are a bit difficult to present in an intuitive manner. Every strip associated with a given image is targeting the same 14-dimensional particle size distribution vector (each dimension is a percentage of ballast remaining at a given sieve size). Due to this, it seemed best to present 14 box plots showing the distribution of guesses across all strips compared to each true value as one of the graphs. The labels on the x axis of the top graph are the true value being guessed (the box plots are also plotted against this value on the y axis). The maximum and minimum guessed percentages are labeled with a dot for each dimension. The bars represent the range of 95% of the distribution while the blue box represents 75% of the distribution. The bottom graph on each figure is a plot of the mean guessed percentage remaining in each dimension vs. the true measured value for a given image.

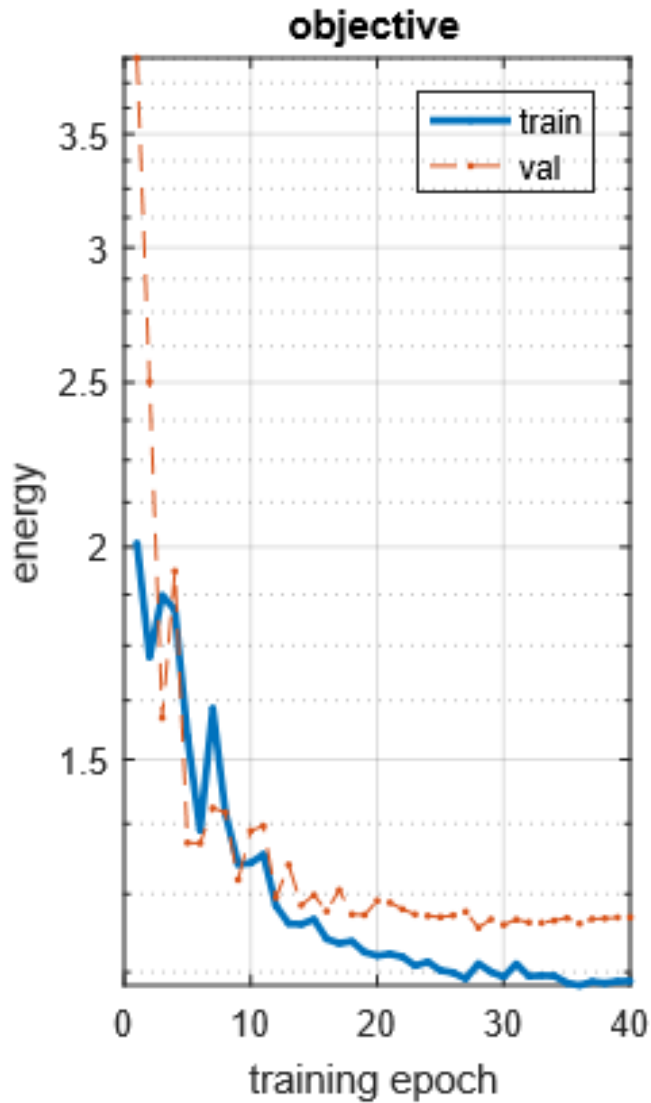


Figure 4.1: Minimization of Objective Function

Figure 4.2 is an example of the results gleaned from training a network on the features from the other 13 images to predict a percentage remaining distribution curve. The results for the rest of the 14 images can be found in the appendix.

The plots serve as a visual means of inspecting whether or not the predicted distribution curves match up to the actual distribution curves and highlight the variance of the predicted values in each dimension.

In addition, the root mean squared error and normalized root mean squared error for all dimensions in the images have been plotted (see an example in figure 4.3). Currently, these charts serve as a means of distinguishing between the results on various images.

Mean squared error in these images is defined as:

$$MSE = \sum_x [(\vec{y}_{predicted} - \vec{y}_{actual})^2] \quad (4.1)$$

where x represents iteration across all the samples and the y vectors represent the actual and predicted particle size distribution values.

The normalized root mean squared error is defined in equation 4.2:

$$NRMSE = \frac{\sqrt{(MSE)}}{\vec{y}_{max} - \vec{y}_{min}} \quad (4.2)$$

It exists to serve as a basis of comparison between the error dimensions, which have significantly different absolute value ranges.

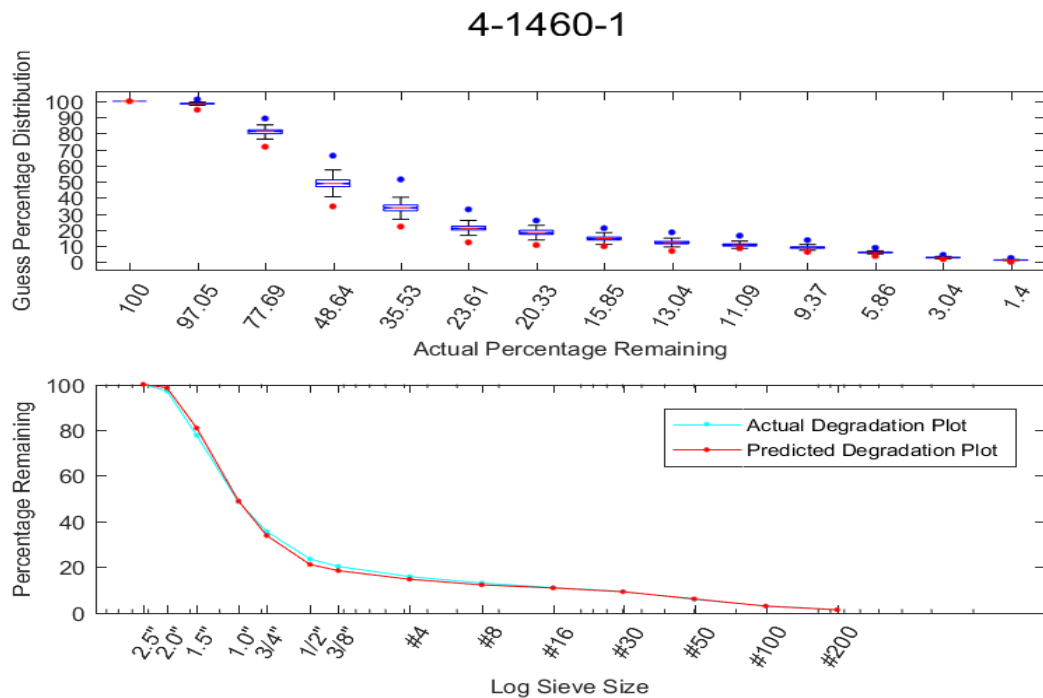


Figure 4.2: Example Cross Sectional Result

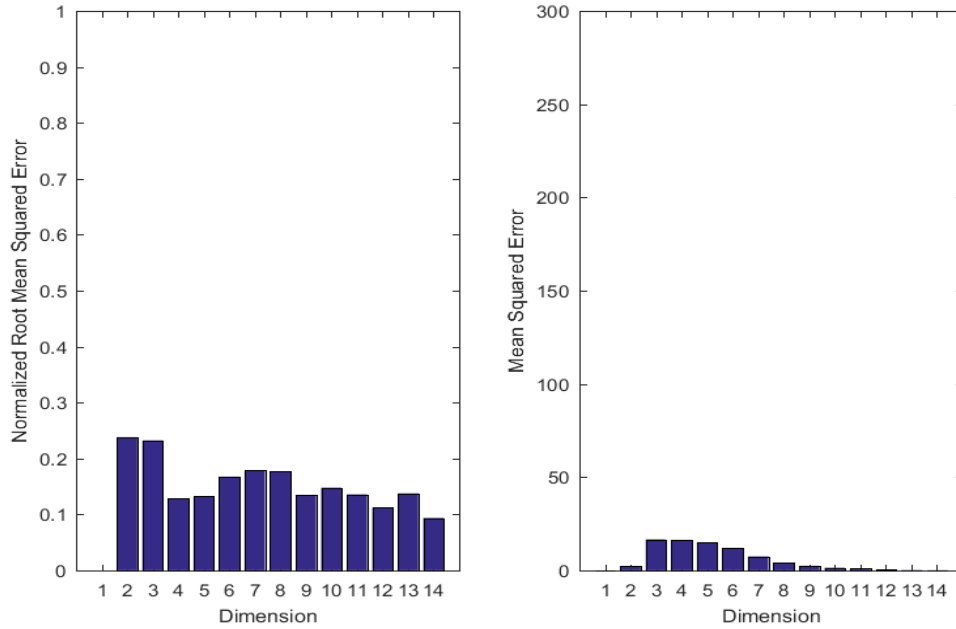


Figure 4.3: Example Error Graphs

The data shown by these bar charts capture the metrics that would be used to compare the prediction results here to the prediction results from other techniques and can be used as an objective metric for prediction accuracy (particularly the MSE; the NRMSE is more useful for seeing what parts of the particle size distribution curve individual classifiers are failing to distinguish). Future attempts at ballast fouling prediction should most likely seek to minimize these metrics in place of the more typical Selig Fouling Index or Percentage Fouling (as these predictions are more general).

It should be noted that there are some issues mapping between these error results and the decision on whether or not to replace the corresponding ballast. There is currently no hard and fast rule for when ballast needs to be replaced. Some literature suggests that it occur when the ballast no longer has a certain permeability to water [18], while some simply use the Selig definition of fouled ballast (an index of 40+). An investigation of the degree of resolution needed when mapping from these results to the eventual decision criterion would be needed for use of this data in the field.

In general, the results of this classification seem to closely match the expected particle size distribution curves. However, as the particle size distri-



bution curves tend to resemble one another, more heavily and lightly fouled samples need to be acquired. If the classifier is equally capable of distinguishing between those samples and maintains a high accuracy in distinguishing between more similar samples, then this approach is well suited to the task of distinguishing levels of ballast degradation.

# CHAPTER 5

## CONCLUSION

A structured, consistent, repeatable method for the prediction of railway ballast fouling is of massive value to the railroad industry. Accurate fouling analysis can inform decisions that directly impact the safety of rail passengers and the reliability of rail transportation. This research has walked through a variety of datasets and machine learning methodologies that try to accomplish that task.

The various datasets and machine learning methods covered highlight the way in which the scope of machine learning problems can change over time. The acquisition of truly representative data is one of the largest barriers to building an accurate machine learning system and this research made significant strides towards figuring out exactly what data was needed to accurately predict ballast degradation. It also showed that the classifiers used to quantify and label ballast fouling are important to the overall accuracy of the system. Image recognition in general is a difficult issue for machine learning algorithms, one that only recently has started to become tractable, and ballast fouling recognition is not significantly different. As in other fields, it appears as though deep learning networks are particularly well-suited to this task.

The initial attempts at tackling this problem suffered from a wide variety of issues including, but not limited to: datasets with no real ground truth labels, classifiers unable to distinguish between various degradation classes, and a lack of training data at certain degradation levels. However, the eventual method settled upon by this research, that of using a CNN to predict a ballast fouling particle size distribution curve, shows significant promise. In particular, the results from that methodology show a close mapping between objective reality and the predictions of the various machine learning algorithms.

However, important caveats remain to be addressed. The cross-sectional

image dataset is a vast improvement over previous datasets, as results on it can be verified, but it still does not contain a representative set of images. A wider variety of images of different fouling indices needs to be acquired. Images of very heavily fouled sections of track that are due to be replaced need to be used for training alongside brand new sections of ballast. Unfortunately, the training set is dominated by relatively clean ballast cross-sections, and no heavily fouled (40+ Selig Fouling Index) examples exist.

Other concerns include the eventual use of this information to make decisions. The availability of ballast fouling prediction algorithms makes little difference if there is no formal way to take those results and make decisions based upon them. This is an issue that would require input from railway operators themselves.

Additionally, there is still room for improvements to the process of particle size distribution prediction. Limitations in hardware resulted in the choice of a fairly simple CNN architecture, but more complex ones would likely result in improvements. Custom architectures based on the eventual input data used by the various railways could yield even more improvements, though CNN architecture design still suffers from a lack of mathematical grounding.

What these problems boil down to at their core are some of the classic difficulties of any machine learning task: the need for huge amounts of representative training data, and the need to properly train an appropriate classifier or predictor on the problem at hand. Once the task and the data are well-defined, optimization should yield huge gains.

These issues present real difficulties and many opportunities for further research, but there is still significant value in this work. It highlights the somewhat exploratory progression of tackling a real-world machine learning problem, and does so in guided way. Building up from simple data, analyzing the problems inherent in that data, and determining what is needed to satisfactorily solve the stated problem (in this case, predicting railroad ballast degradation from an image) are the principal challenges of machine learning and automation in a nutshell.

The results in this thesis demonstrate that this task is possible, and can hopefully offer some insight into how to construct automated ballast degradation analysis tools. Specific applications of these techniques will require trained networks and tuned datasets, but the general approach should remain the same.

## REFERENCES

- [1] H. Huang, E. Tutumluer, and W. Dombrow, “Laboratory characterization of fouled railroad ballast behavior,” *Transportation Research Record: Journal of the Transportation Research Board*, no. 2117, pp. 93–101, 2009.
- [2] E. T. Selig and J. M. Waters, *Track Geotechnology and Substructure Management*. Thomas Telford, 1994.
- [3] M. Moaveni, Y. Qian, H. Boler, D. Mishra, and E. Tutumluer, “Investigation of ballast degradation and fouling trends using image analysis,” in *Proceedings of the Second International Conference on Railway Technology: Research, Development and Maintenance*, 2014.
- [4] M. Moaveni, S. Wang, J. Hart, E. Tutumluer, and N. Ahuja, “Evaluation of aggregate size and shape by means of segmentation techniques and aggregate image processing algorithms,” *Transportation Research Record: Journal of the Transportation Research Board*, no. 2335, pp. 50–59, 2013.
- [5] W. Lim, M. Brough, and S. Middleton, “Prediction of ballast return from high output ballast cleaners (HOBC),” Scott Wilson Pavement Engineering. CiteseerX, 2010.
- [6] L. Devroye, “Sample-based non-uniform random variate generation,” in *Proceedings of the 18th Conference on Winter Simulation*. ACM, 1986, pp. 260–265.
- [7] *Standard Test Method for Sieve Analysis of Fine and Coarse Aggregates*, ASTM International Std. C136, 2001.
- [8] B. Delay, M. Moaveni, J. Hart, P. Sharpe, and E. Tutumluer, “Evaluation of degradation conditions in automated ballast sampler using machine vision and machine learning techniques,” in *ASCE Geotechnical Frontiers Proceedings*, 2017, written in 2016, to be presented in 2017.
- [9] M. A. Nielsen, *Neural Networks and Deep Learning*. Determination Press, 2015.

- [10] D. Harris and S. Harris, *Digital Design and Computer Architecture, Second Edition*, 2nd ed. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 2012.
- [11] D. Reynolds, “Gaussian mixture models,” *Encyclopedia of Biometrics*, pp. 827–832, 2015.
- [12] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, A. C. Berg, and L. Fei-Fei, “ImageNet Large Scale Visual Recognition Challenge,” *International Journal of Computer Vision (IJCV)*, vol. 115, no. 3, pp. 211–252, 2015.
- [13] C.-K. Shie, C.-H. Chuang, C.-N. Chou, M.-H. Wu, and E. Y. Chang, “Transfer representation learning for medical image analysis,” in *2015 37th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)*. IEEE, 2015, pp. 711–714.
- [14] J. Yosinski, J. Clune, Y. Bengio, and H. Lipson, “How transferable are features in deep neural networks?” *CoRR*, vol. abs/1411.1792, 2014. [Online]. Available: <http://arxiv.org/abs/1411.1792>
- [15] D. E. Rumelhart, G. E. Hinton, and R. J. Williams, “Learning representations by back-propagating errors,” *Cognitive Modeling*, vol. 5, no. 3, p. 1, 1988.
- [16] D. Wei, B. Zhou, A. Torralba, and W. T. Freeman, “mNeuron: A Matlab plugin to visualize neurons from deep models,” 2015, [Online; accessed Oct 2, 2016]. [Online]. Available: [http://vision03.csail.mit.edu/cnn\\_art/data/single\\_layer.png](http://vision03.csail.mit.edu/cnn_art/data/single_layer.png)
- [17] C. Sammut and G. I. Webb, Eds., *Encyclopedia of Machine Learning*. Boston, MA: Springer US, 2010.
- [18] N. Tennakoon, B. Indraratna, C. Rujikiatkamjorn, and S. Nimbalkar, “Assessment of ballast fouling and its implications on track drainage,” in *New Zealand Conference on Geomechanics: Ground Engineering in a Changing World*, 2012.

# APPENDIX A

## CROSS SECTIONAL BALLAST IMAGE RESULTS

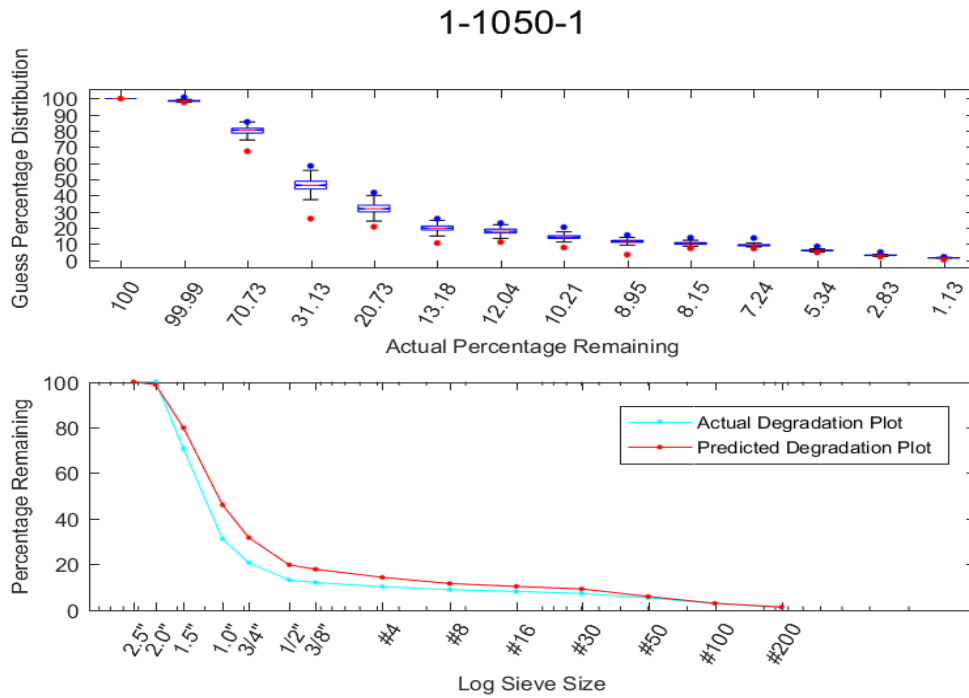


Figure A.1: 1-1050-1 Particle Size Distribution Characteristics

### 1-1050-1

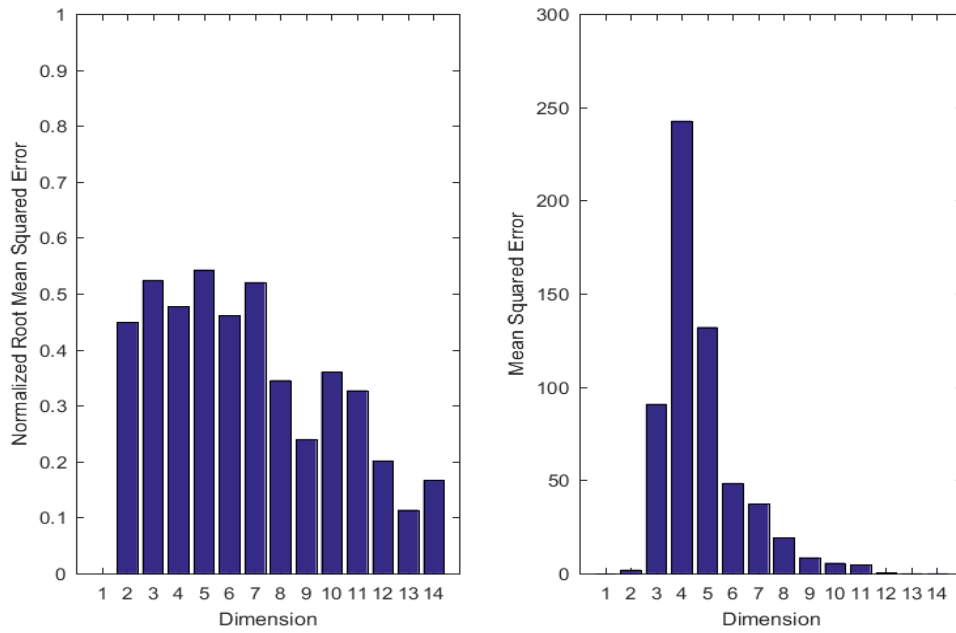


Figure A.2: 1-1050-1 Error Characteristics

### 1-1050-2

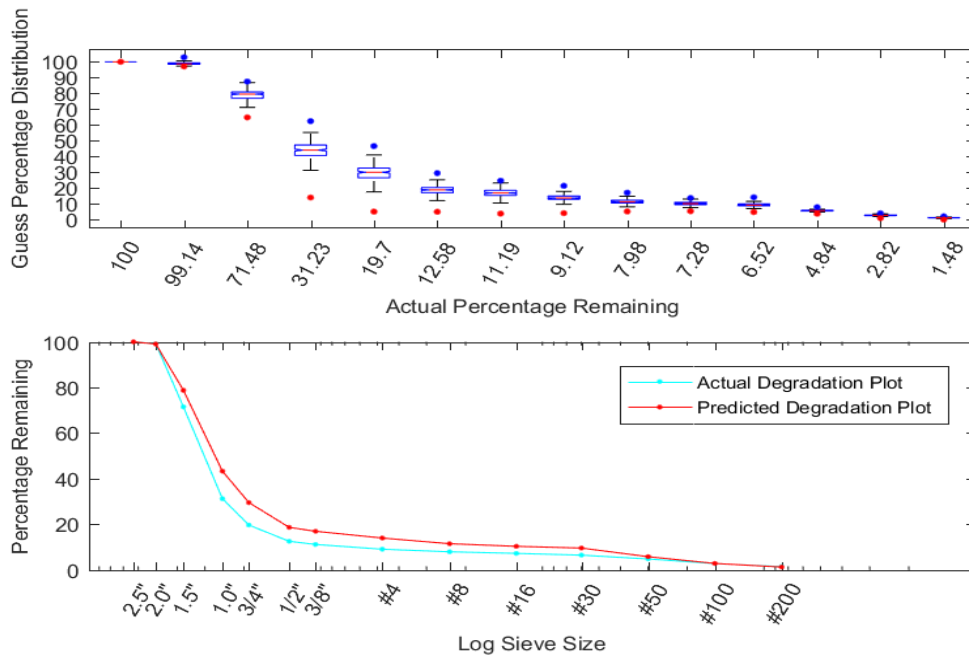


Figure A.3: 1-1050-2 Particle Size Distribution Characteristics

### 1-1050-2

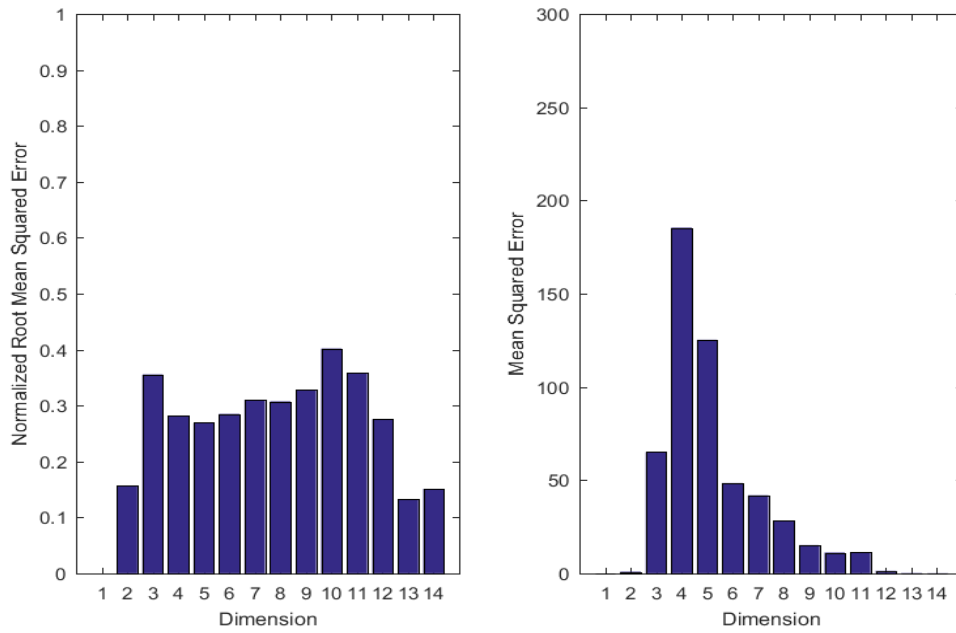


Figure A.4: 1-1050-2 Error Characteristics

### 2-1308-1

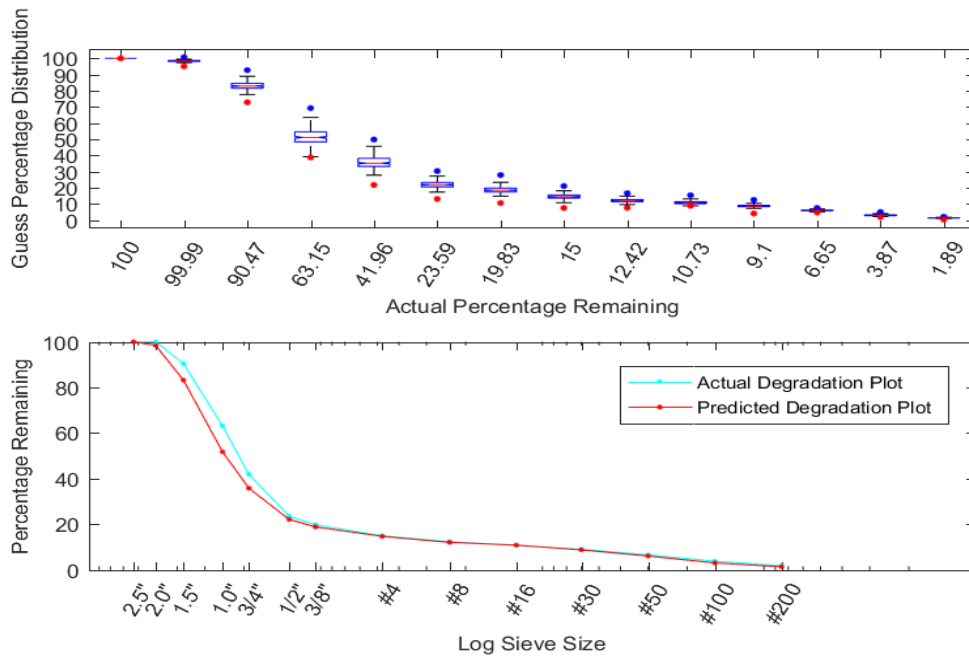


Figure A.5: 2-1308-1 Particle Size Distribution Characteristics



### 2-1308-1

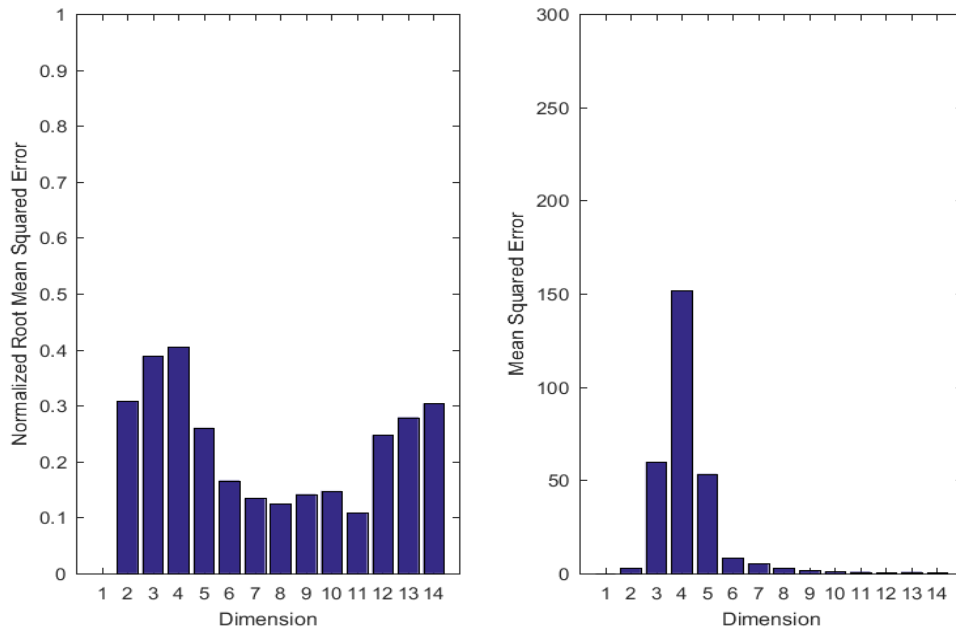


Figure A.6: 2-1308-1 Error Characteristics

### 2-1308-2

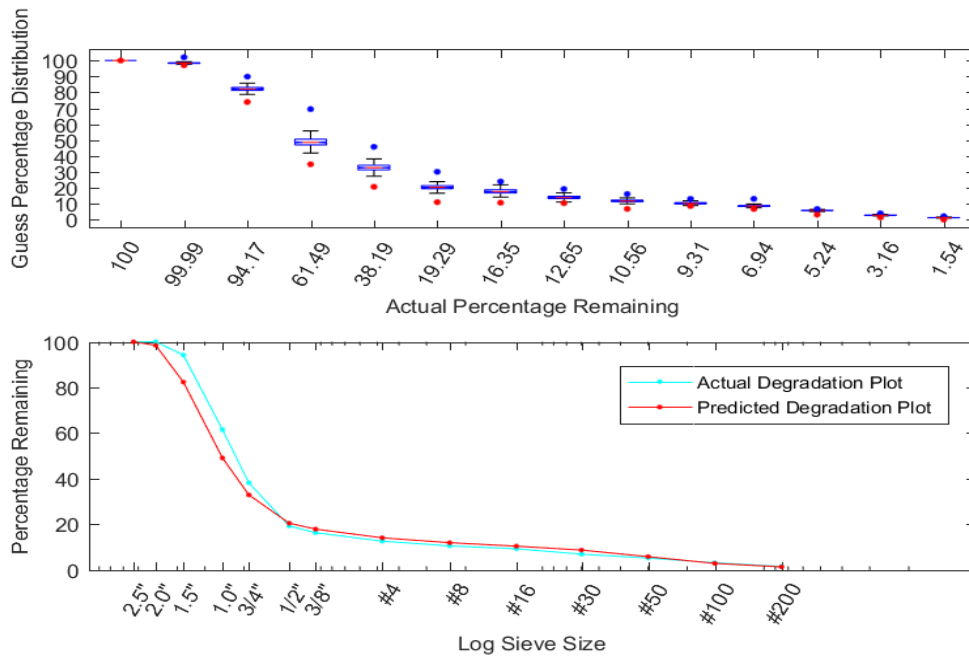


Figure A.7: 2-1308-2 Particle Size Distribution Characteristics

## 2-1308-2

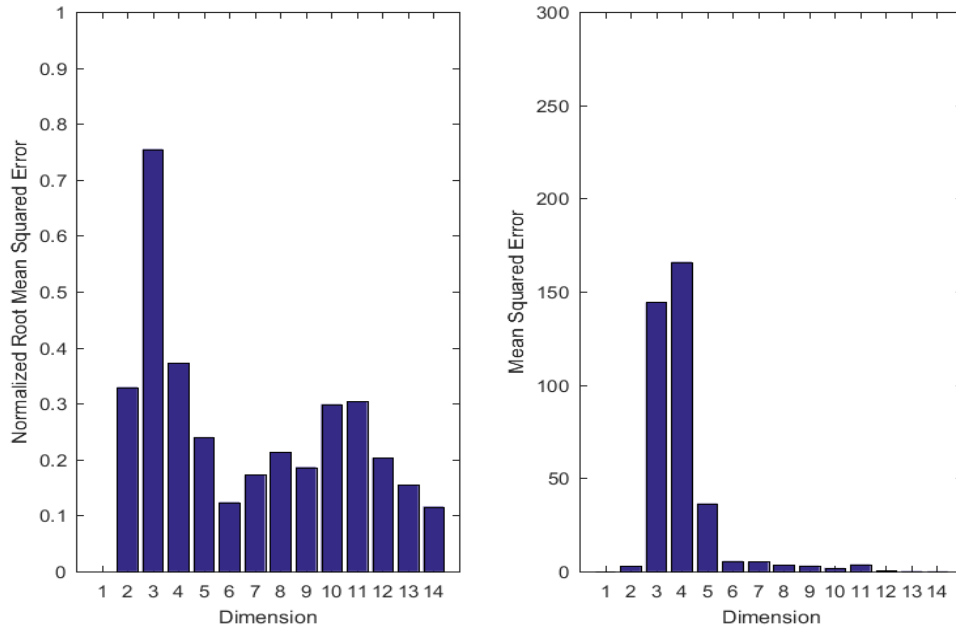


Figure A.8: 2-1308-2 Error Characteristics

## 3-1354-C1-noball

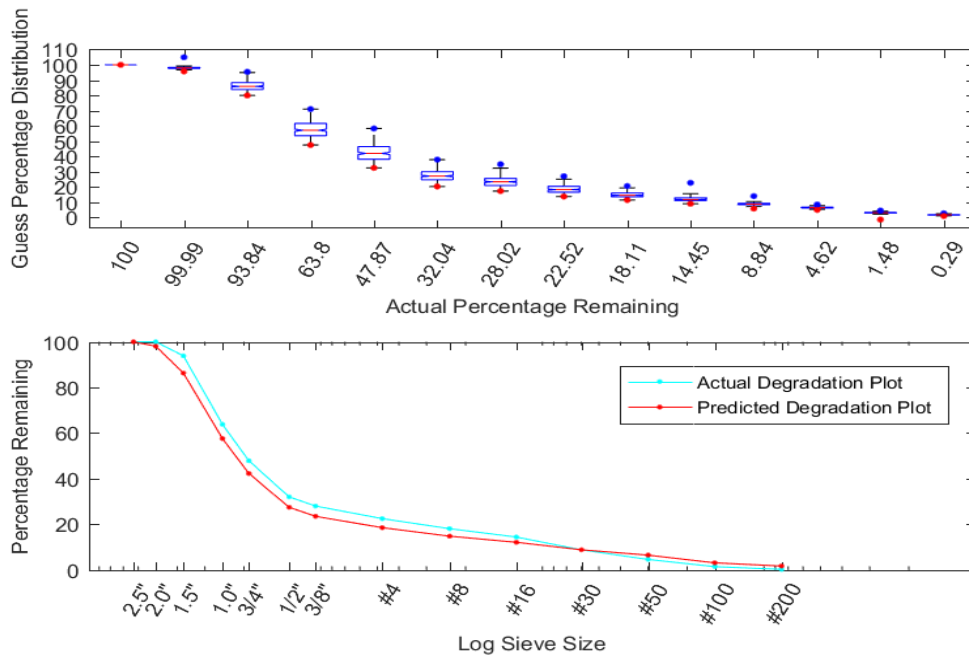


Figure A.9: 3-1354-C1-noball Particle Size Distribution Characteristics

### 3-1354-C1-noball

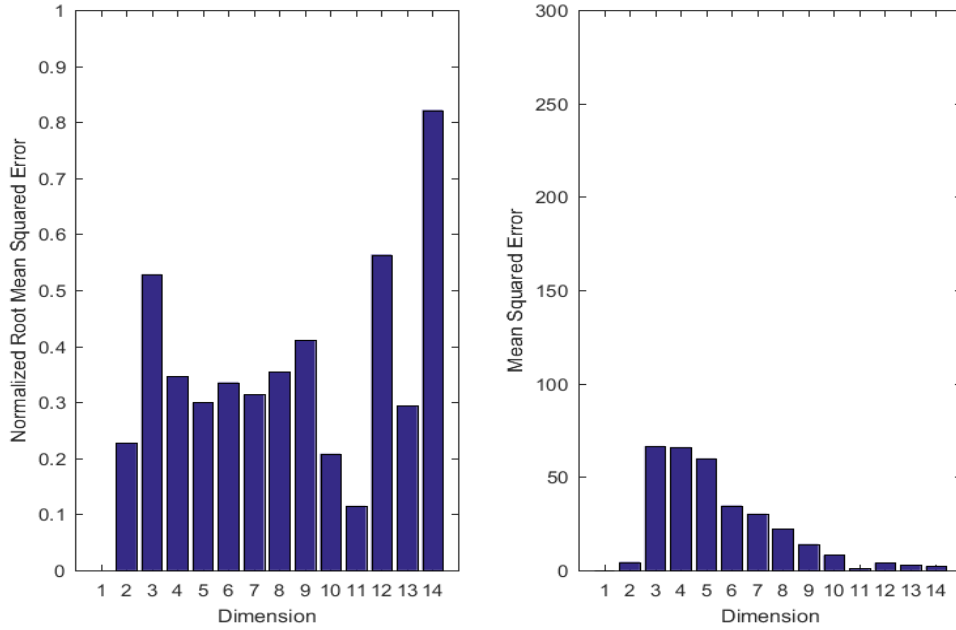


Figure A.10: 3-1354-C1-noball Error Characteristics

### 3-1354-C1-Wall2-Panoramic

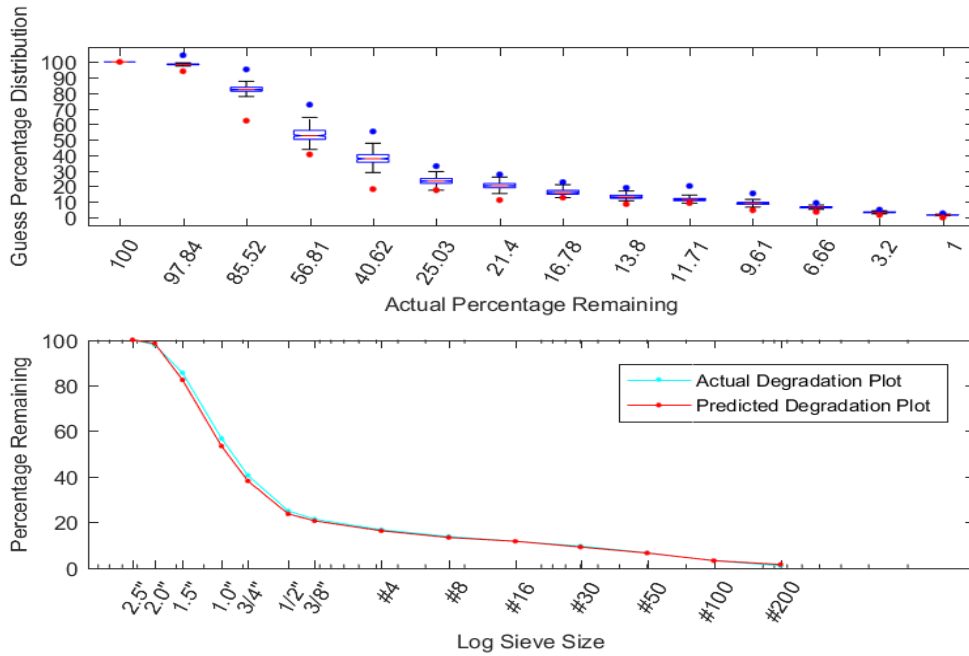


Figure A.11: 3-1354-C1-Wall2-Panoramic Particle Size Distribution Characteristics

### 3-1354-C1-Wall2-Panoramic

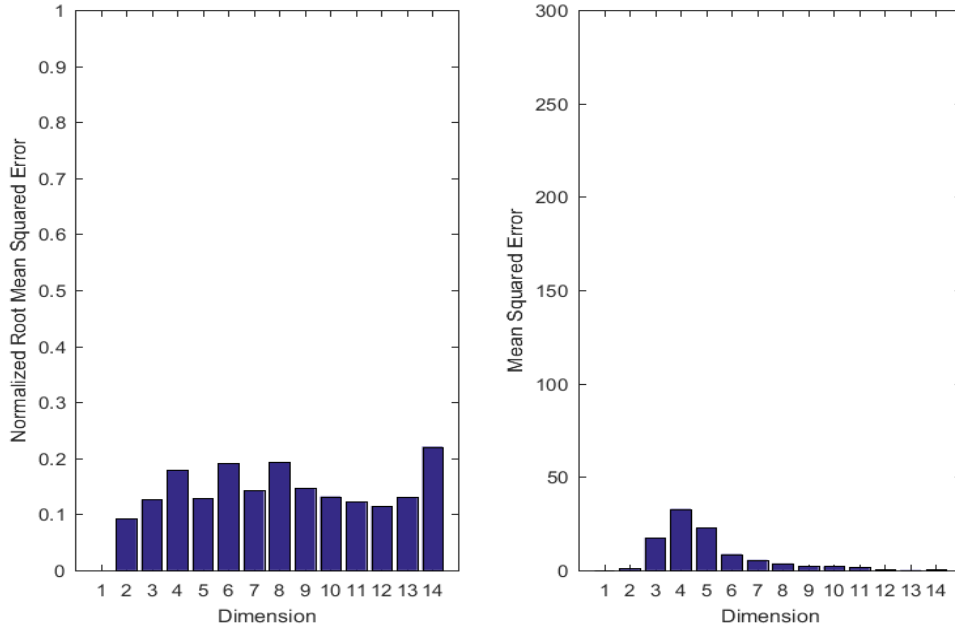


Figure A.12: 3-1354-C1-Wall2-Panoramic Error Characteristics

### 3-1354-C2-Wall2-Panoramic

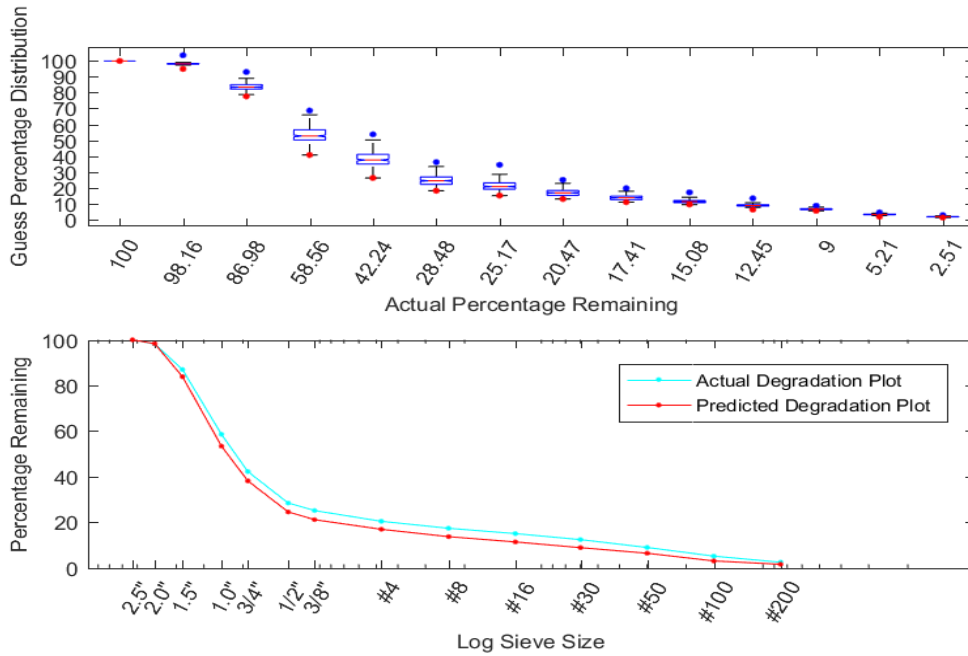


Figure A.13: 3-1354-C2-Wall2-Panoramic Particle Size Distribution Characteristics

### 3-1354-C2-Wall2-Panoramic

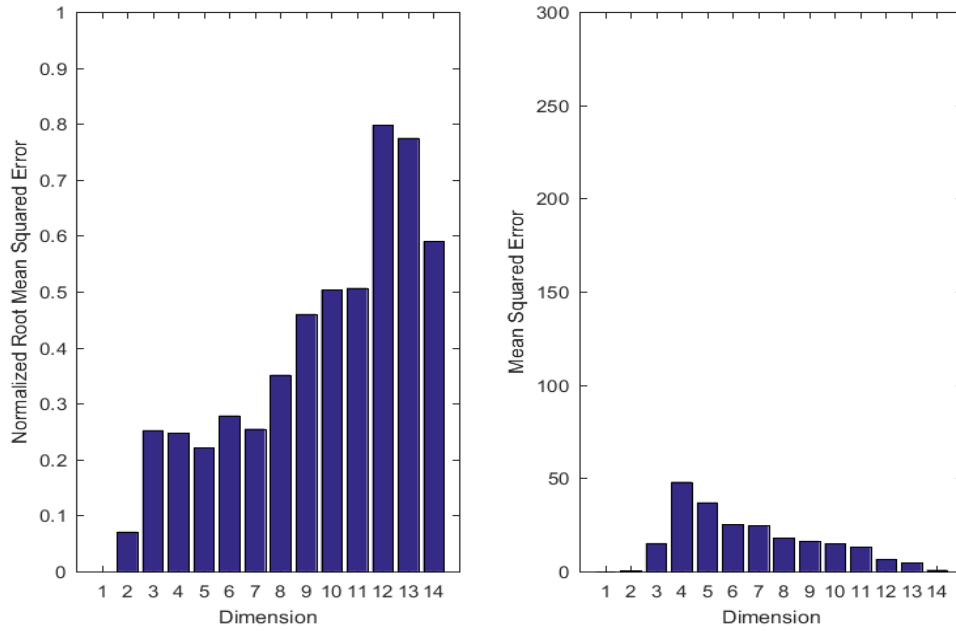


Figure A.14: 3-1354-C2-Wall2-Panoramic Error Characteristics

### 3-1354-I1-noball

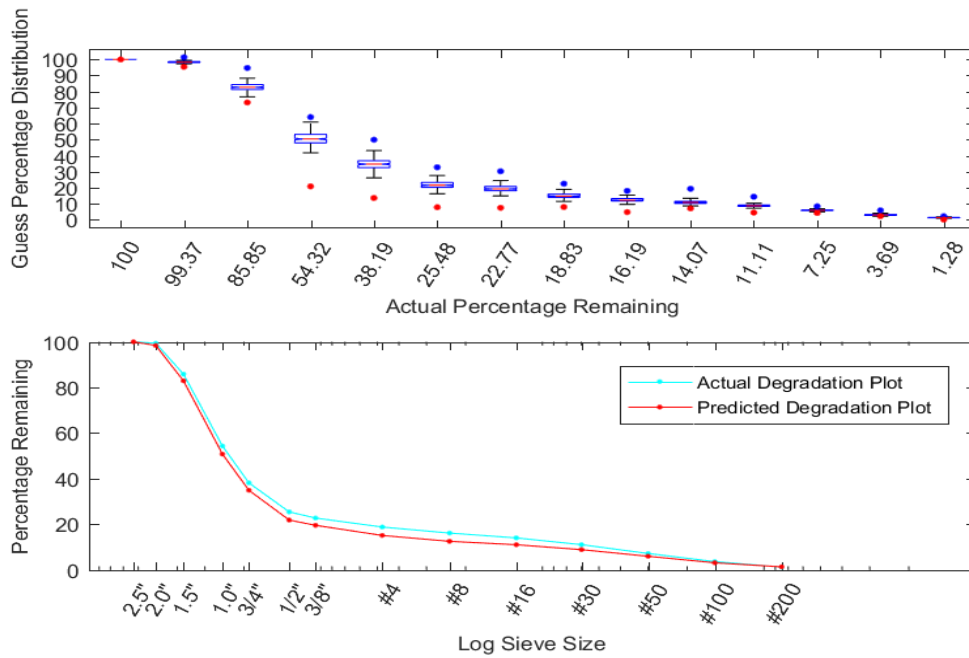


Figure A.15: 3-1354-I1-noball Particle Size Distribution Characteristics

### 3-1354-I1-noball

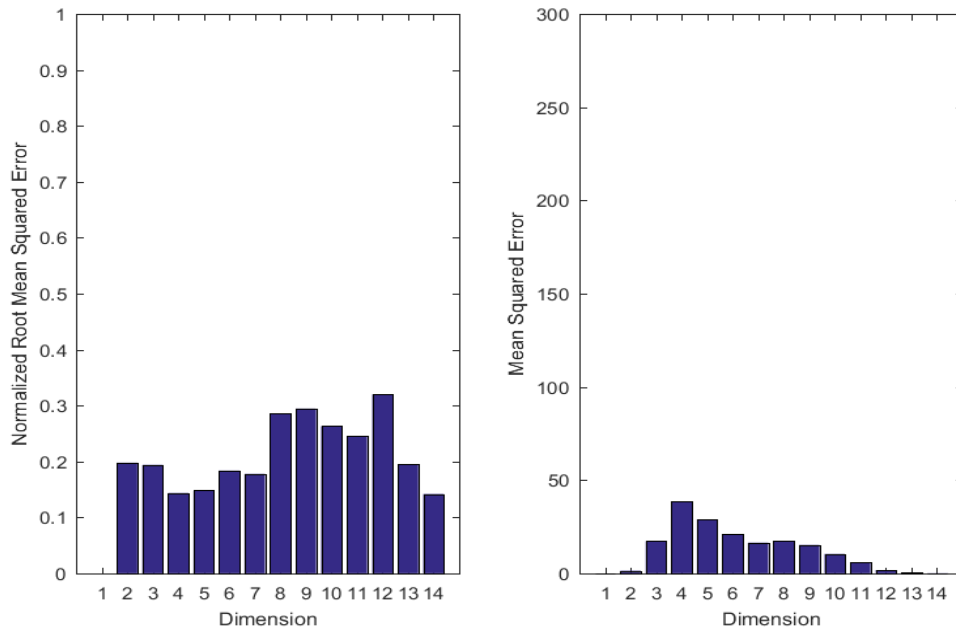


Figure A.16: 3-1354-I1-noball Error Characteristics

### 3-1396-1

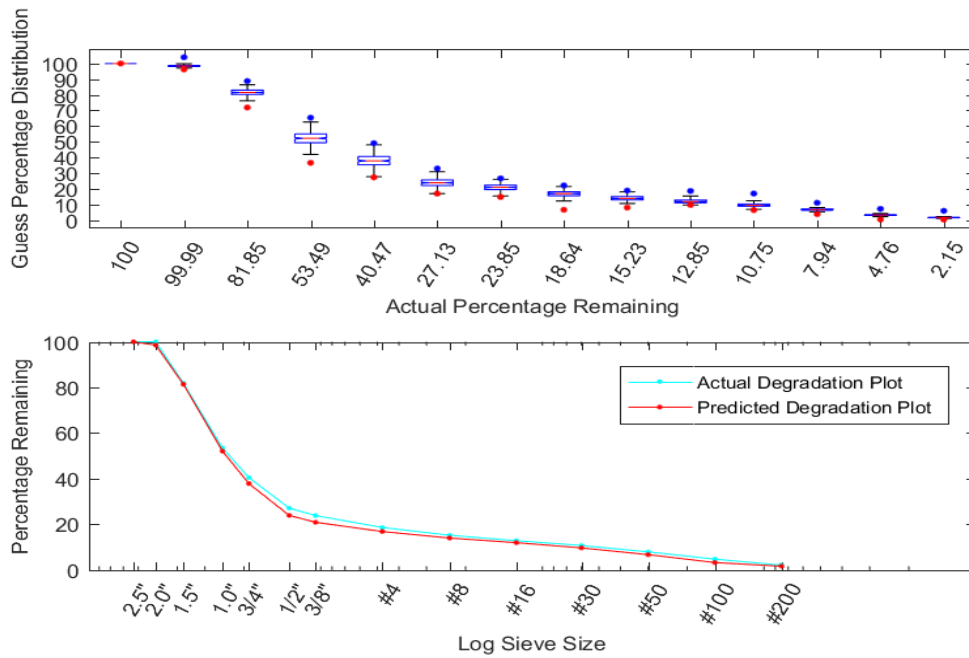


Figure A.17: 3-1396-1 Particle Size Distribution Characteristics

### 3-1396-1

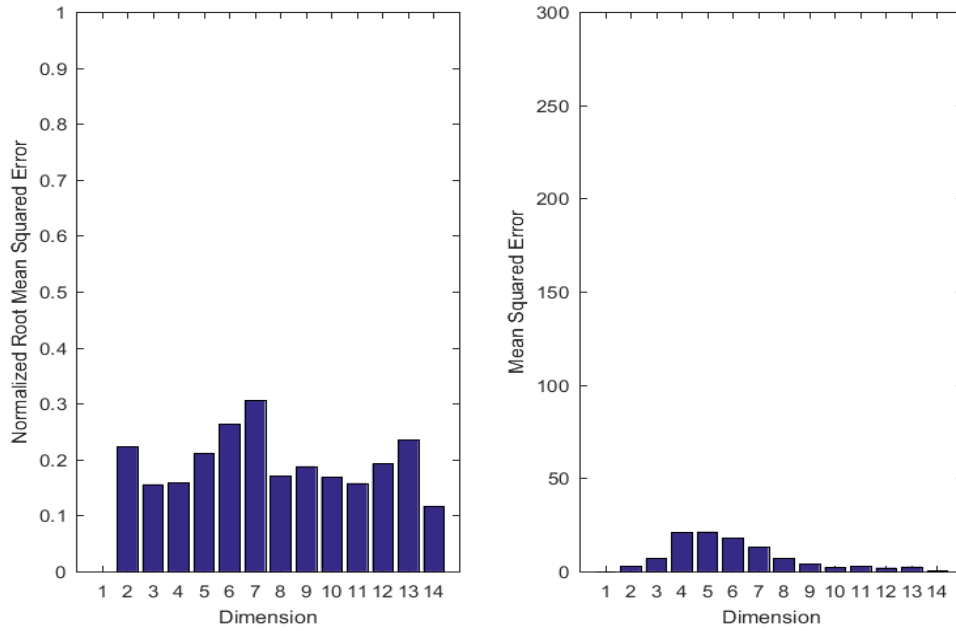


Figure A.18: 3-1396-1 Error Characteristics

### 3-1396-2

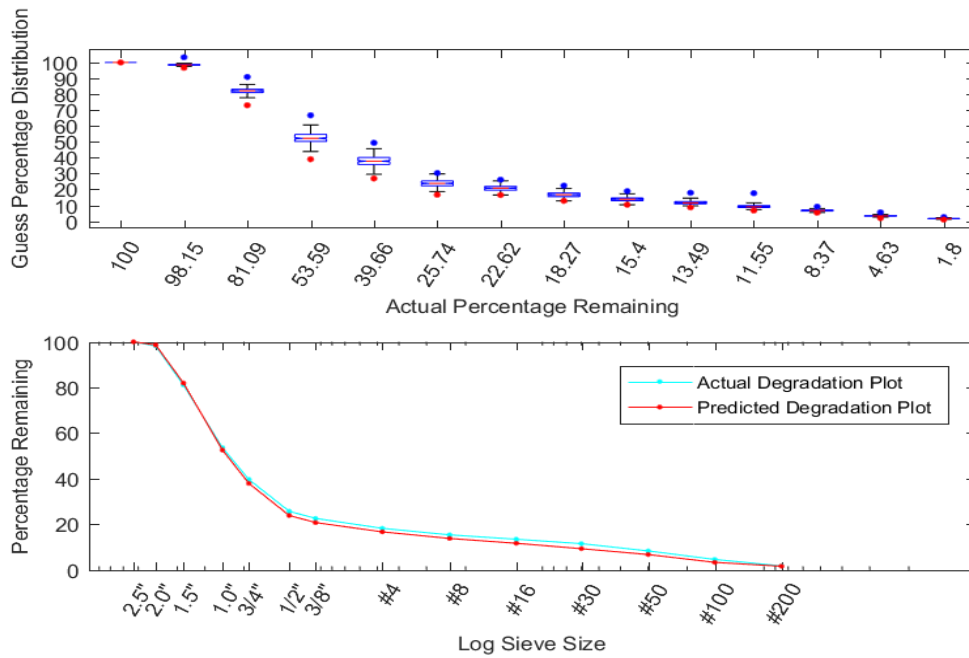


Figure A.19: 3-1396-2 Particle Size Distribution Characteristics

### 3-1396-2

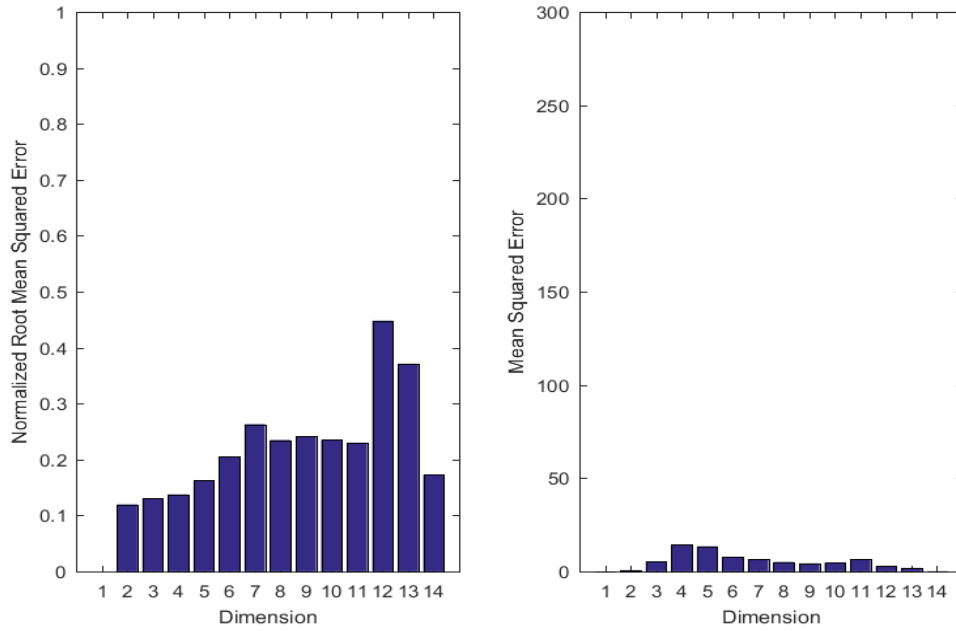


Figure A.20: 3-1396-2 Error Characteristics

### 4-1460-1

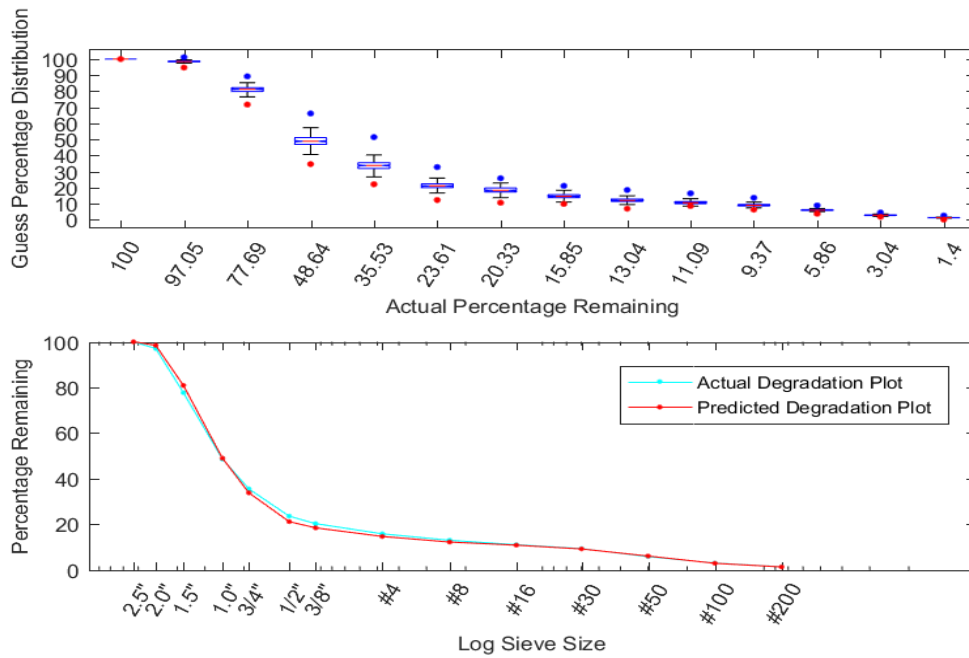


Figure A.21: 4-1460-1 Particle Size Distribution Characteristics



### 4-1460-1

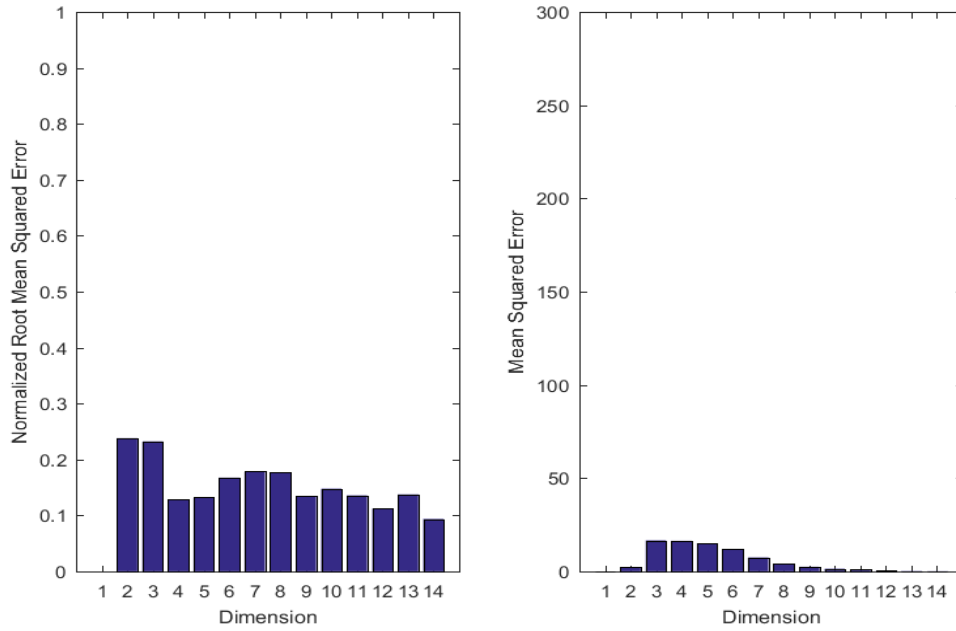


Figure A.22: 4-1460-1 Error Characteristics

### 4-1460-2

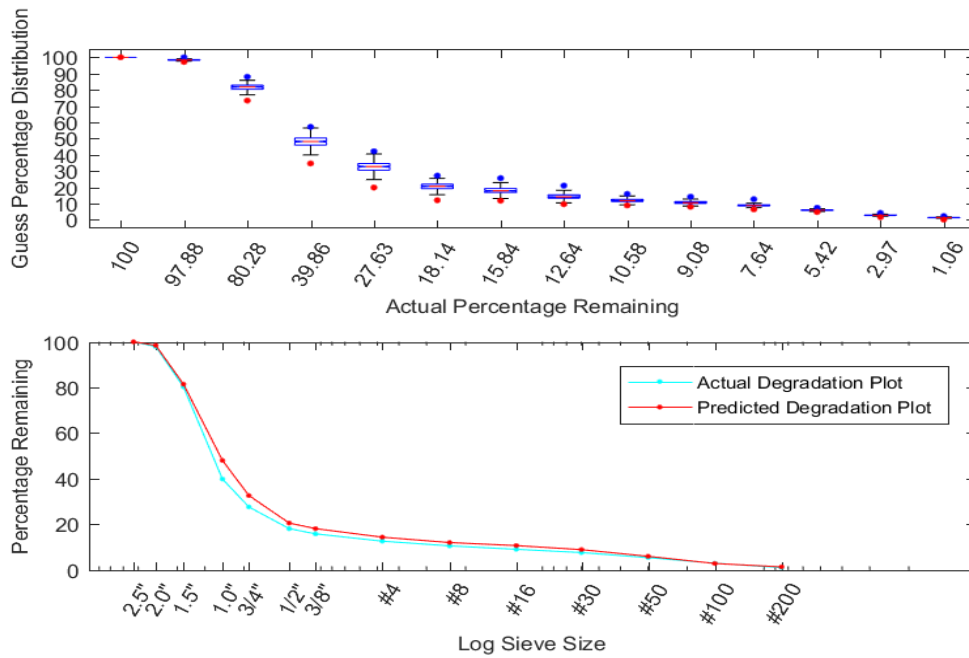


Figure A.23: 4-1460-2 Particle Size Distribution Characteristics

### 4-1460-2

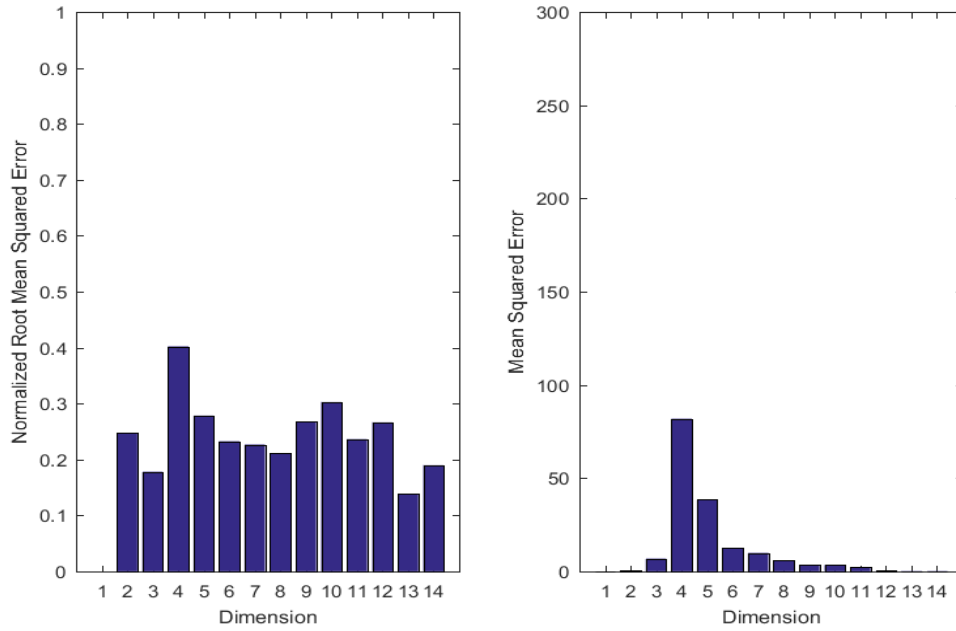


Figure A.24: 4-1460-2 Error Characteristics

### 5-1557-1

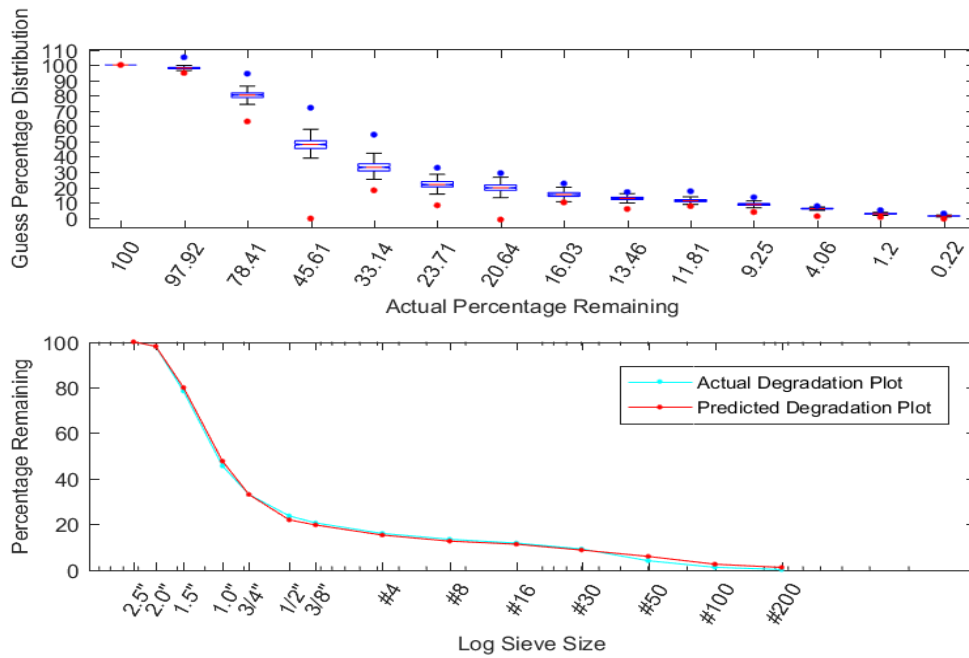


Figure A.25: 5-1557-1 Particle Size Distribution Characteristics

### 5-1557-1

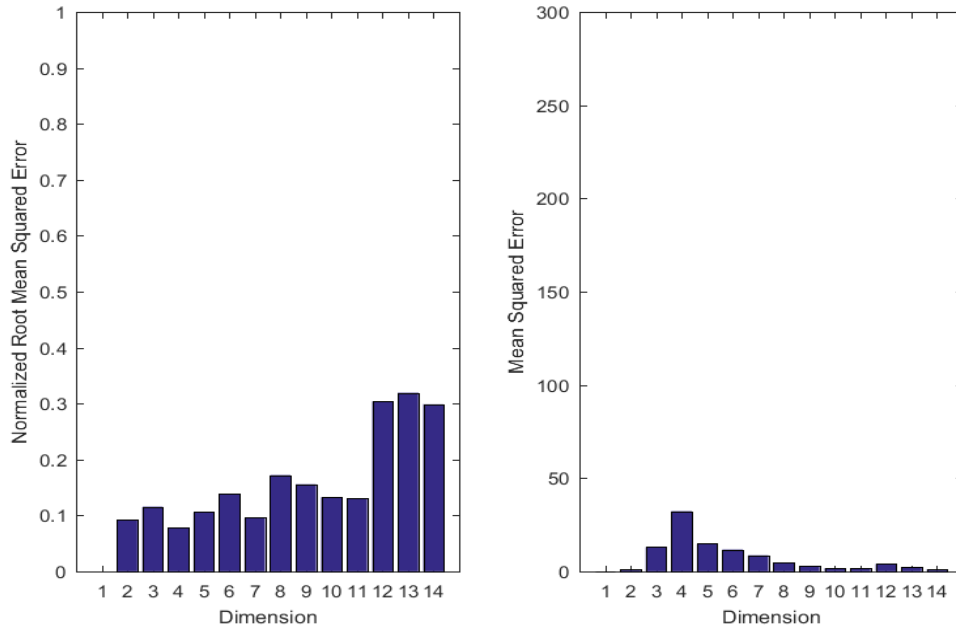


Figure A.26: 5-1557-1 Error Characteristics

### 5-1557-2

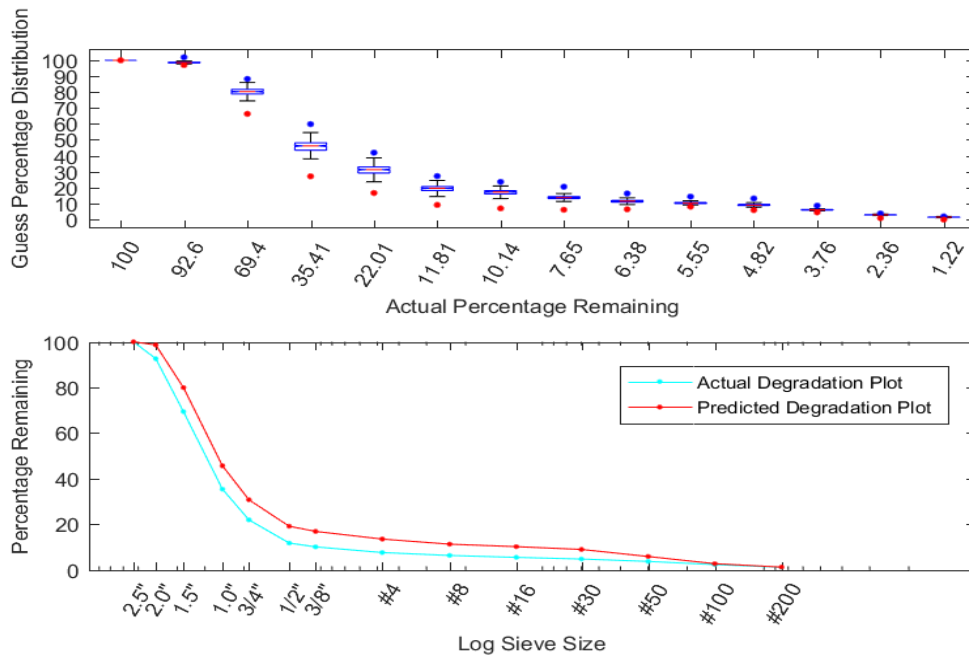


Figure A.27: 5-1557-2 Particle Size Distribution Characteristics

### 5-1557-2

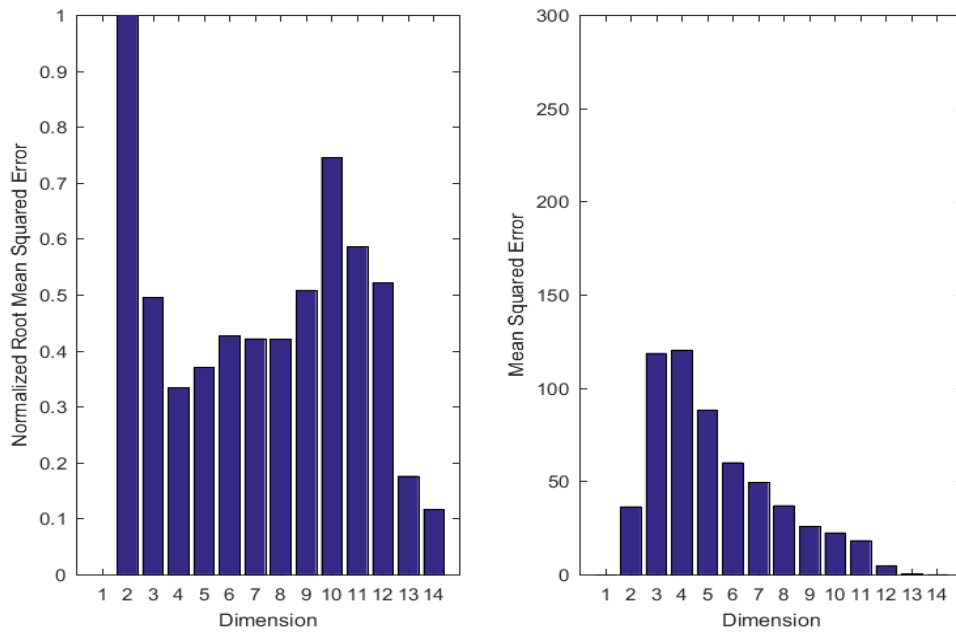


Figure A.28: 5-1557-2 Error Characteristics