

ESTIMATING LENGTH STATISTICS OF AGGREGATE  
FRIED POTATO PRODUCT VIA ELECTROMAGNETIC  
RADIATION ATTENUATION

by  
Jesse Lovitt

A thesis  
submitted in partial fulfillment  
of the requirements for the degree of  
Master of Science in Computer Science  
Boise State University

December 2016

© 2016  
Jesse Lovitt  
ALL RIGHTS RESERVED

BOISE STATE UNIVERSITY GRADUATE COLLEGE

**DEFENSE COMMITTEE AND FINAL READING APPROVALS**

of the thesis submitted by

Jesse Lovitt

Thesis Title: Estimating Length Statistics of Aggregate Fried Potato Product via Electromagnetic Radiation Attenuation

Date of Final Oral Examination: 27 October 2016

The following individuals read and discussed the thesis submitted by student Jesse Lovitt, and they evaluated the presentation and response to questions during the final oral examination. They found that the student passed the final oral examination.

Tim Andersen Ph.D.

Chair, Supervisory Committee

Edoardo Serra Ph.D.

Member, Supervisory Committee

Aaron Westcott

Member, Supervisory Committee

The final reading approval of the thesis was granted by Tim Andersen Ph.D., Chair of the Supervisory Committee. The thesis was approved by the Graduate College.

# ACKNOWLEDGMENTS

This work was made possible by the generosity of the Simplot corporation both in financial resources and access to valuable data as well as the Idaho National Laboratory whose generous computing resources are greatly appreciated. Without the valuable insights of the committee members Tim Andersen, Aaron Westcott, and Edoardo Serra this effort would not have been nearly as rewarding. Thanks must also go to my dedicated family who made uncountable sacrifices for me to get here.

This research made use of the resources of the High Performance Computing Center at Idaho National Laboratory, which is supported by the Office of Nuclear Energy of the U.S. Department of Energy under Contract No. DEAC0705ID14517.

# ABSTRACT

This work investigates the feasibility of using non-destructive testing, in particular radiation tomography, to recover length statistics from aggregates of fried batonnet cut potato. Non-destructive testing comprises a variety of useful techniques for determining properties of an object that might otherwise require altering or destroying the object physically. Tomography is a common form of non-destructive testing used primarily to infer properties internal to an object. This process involves exposing the object of interest to radiation and detecting the quantity of radiation energy that penetrates the object, usually resulting in a grey scale image.

To do this, an artificial data pipeline is developed in order to obtain annotated examples. This pipeline allows faster data collection than can be done in a real production environment coupled with the ability to control all aspects of the resulting images. Additionally, these examples are used to train a convolutional neural network, a widely successful machine learning algorithm for image processing. The network learns a relation between the images and the length estimates and can then be used to provide length estimates on novel examples.

Results show that with careful preparation and enough expected variation in the product being inspected, the image resulting from radiation tomography contains enough information to recover estimates of the lengths of the product with significantly less expected error than a naive baseline.

# TABLE OF CONTENTS

ACKNOWLEDGMENTS .....	iv
ABSTRACT .....	v
LIST OF TABLES .....	x
LIST OF FIGURES .....	xi
LIST OF ABBREVIATIONS .....	xiv
<b>1 Introduction</b> .....	<b>1</b>
1.1 Non-Destructive Testing .....	1
1.2 Radiation Tomography .....	2
1.3 Deep Convolutional Nets .....	4
1.4 Thesis Statement .....	5
<b>2 Related Work</b> .....	<b>6</b>
<b>3 Data Generation</b> .....	<b>8</b>
3.1 Artificial Image Generation .....	8
3.2 Distribution Modeling .....	10
3.2.1 Width Distribution Modeling .....	10
3.2.2 Density Distribution Modeling .....	11
3.2.3 Length Distribution Modeling .....	12
3.3 Distribution Sampling .....	15
3.3.1 Historically Likely Samples .....	16

3.3.2	Historically Unlikely Samples .....	17
3.3.3	Sampling Lengths from Bin Level Distribution .....	19
3.4	Mesh Generation .....	22
3.5	Simulation .....	23
3.6	Exposure .....	24
3.6.1	Radiation Attenuation .....	25
3.6.2	Detection Sensitivity .....	27
3.6.3	Spatial Parameters .....	30
3.7	Image Based Validation .....	31
3.7.1	Color Histograms .....	32
3.7.2	Linear Binary Pattern Histograms .....	32
3.7.3	Measures of Central Tendency .....	34
3.8	Image Staging .....	36
3.8.1	Orientation and Cropping .....	36
<b>4</b>	<b>Network, Objective, and Feature Selection .....</b>	<b>39</b>
4.1	Feature Selection .....	39
4.1.1	Motivation for Hand Made Features .....	39
4.1.2	Edge Features .....	41
4.1.3	Pattern Features .....	42
4.2	Objective Selection .....	43
4.2.1	Network Outputs .....	43
4.2.2	Objective Function .....	45
4.3	Model Architecture Selection .....	47
4.4	MetaModel .....	49

<b>5</b>	<b>Performance Evaluation</b>	51
5.1	Data Sets	51
5.1.1	Without Realism Factors	51
5.1.2	With Realism Factors	51
5.2	Baselines	52
5.2.1	Immediately Feasible	53
5.2.2	Convolutional Neural Network Alternatives	54
5.3	Evaluation	56
5.3.1	Realism	57
5.3.2	Historical Accuracy	57
5.3.3	Findings	58
<b>6</b>	<b>Conclusions</b>	59
6.1	Current State and Future Directions	59
6.2	Improvements To Modeling	60
6.3	Improvements To Preparation	61
6.3.1	Image Stretch	61
6.3.2	Image Resolution	62
6.3.3	Overexposure	64
6.3.4	Bag Orientation	65
6.3.5	Product Dispersion	65
6.3.6	Data Collection	66
6.3.7	The Value in Sampling	68
	<b>REFERENCES</b>	69

<b>A Model Architecture Details</b> .....	72
<b>B Model Performance Details</b> .....	74
<b>C Artifacts</b> .....	77

# LIST OF TABLES

3.1	Weights of the 5 Gaussian mixture model components used to model the bag length distributions . . . . .	15
3.2	Mean bin counts for each of the 5 Gaussian mixture model components.	15
3.3	Linear attenuation coefficients $\mu$ and Hounsfield unit values (HU) for relevant substances. . . . .	27
5.1	Details of the differences in parameters used between the ideal and realistic training sets. . . . .	52
A.1	Details of the network architecture used for training on both data sets with realism and the mixed data set without realism. . . . .	72
A.2	Details of the network architecture used for training on the historical data set without realism. . . . .	73
B.1	The types of outputs and set sizes for each training context. . . . .	75
B.2	Performance of models against artificial test images by the type of data they were trained on. See the top of this section for a description. . . . .	75
B.3	Performance of models against real test images by the type of data they were trained on. See the top of this section for a description. . . . .	76

# LIST OF FIGURES

3.1	A histogram of historical product density. . . . .	12
3.2	Projections of the 5 bin historical data onto the first 3 of 4 PCA components and the last 3 of 4 PCA components respectively. Note that the axes are each some linear combination of the original 5 bins and thus are not easily interpretable. . . . .	14
3.3	Histograms of the distribution of the historical data used for modeling relative to 3 aspects of the target specification. . . . .	16
3.4	Comparison among the distributions of historically likely and both strategies of historically unlikely distributions. The points plotted are actual example samples. Historically likely samples are blue, scaled non-historical samples are green, and iterative non-historical samples are orange. . . . .	19
3.5	An example of the final orientation of a set of meshes after simulation. .	25
3.6	An example of the overexposure effect on the left compared with a normal example on the left. . . . .	28
3.7	The relationship between collected flux and final image output value used in the artificial exposure process. The dotted line shows two standard deviations of randomness added to the relationship to capture the overexposure effect in the real images. . . . .	29
3.8	An example of the noise present in the real images. . . . .	30
3.9	Tomographic image output value histograms for real and artificial images.	33
3.10	Local binary patterns average histograms for a set of real and artificial images. . . . .	34

3.11	Per pixel image statistics. From left to right: mean, variance, skew. From top to bottom: artificial, real, difference . . . . .	35
4.1	Cross correlation energy of several Gabor edge filters. These energies are obtained by convolving the filter over whole tomography images and computing the sum of the pointwise energies. This is done for several different image sets each representing a different mean product length. . . . .	40
4.2	Count of different Local Binary Pattern features. These counts are obtained by computing the LBP at each point in each whole tomography image and computing the sum for each pattern type. This is done for several different image sets each representing a different mean product length. . . . .	41
4.3	An image example input into the network. The images in order are: top row: original, fine DOG, broad DOG, Laplacian, Sobel. bottom row: LBP features 0-5. Note that the augmented image has been normalized by subtracting the training set mean and dividing by the training set variance. Each channel has been normalized to cover the full spectrum for the purposes of print. . . . .	44
6.1	An image example of image stretch likely due to conveyor belt slowing. The image on the left is a normal real image. On the right is a real image displaying stretch. . . . .	62

6.2 An example of large scale variation in real images. The images on the left are the same real image. On the right are an artificial one. The blurred version of these images on the bottom illustrates the brightness variations in the real image compared to the lesser variation in the artificial one..... 67

# LIST OF ABBREVIATIONS

**CNN** – Convolutional Neural Network

**DOG** – Difference of Gaussian

**LOG** – Laplacian of Gaussian

**LBP** – Local Binary Pattern

**PDF** – Probability Density Function

**CDF** – Cumulative Distribution Function

**PCA** – Principal Component Analysis

**GMM** – Gaussian Mixture Model

**LAC** – Linear Attenuation Coefficient

**ReLU** – Rectified Linear Unit

**IBR** – Inspection Band Radiation

# CHAPTER 1

## INTRODUCTION

This work will focus on radiation tomography of agricultural products, specifically fried batonnet cut potato referred to as "cuts" individually and "product" in aggregate. Additionally the focus is on inspecting product only after processing and packaging. To facilitate the investigation examples of product measurements and images of packaged product from an operating production line have been used.

### 1.1 Non-Destructive Testing

Non-destructive testing aims at determining physical or chemical properties of an object or material without altering the material or object. This is particularly useful in areas where altering the object or material would render it less useful or useless. A common and well known example is the use of X-rays to detect cracks or breaks in the internal structure of a bone. Physical inspection would require damaging tissue to access the bone inside.

The use of Non-destructive testing goes well beyond just the detection of broken bones. It is used in a wide variety of applications including weld verification, mechanical part fabrication, medical imaging, archaeology, mining, and food services. In the agricultural and food industries, it is used to identify the properties of packaged product without compromising the packaging or to determine the quality of the inside of a whole food product without damaging it.

Non-destructive techniques include testing a sample's density, firmness, vibrational characteristics, X-ray and gamma ray transmission, optical reflectance and

transmission, electrical properties, and nuclear magnetic resonance. [5]

## 1.2 Radiation Tomography

Radiation tomography leverages the fact that high energy photons are capable of passing through different materials with a relatively fixed probability based on the material and the photon energy. Radiation tomography for food products generally uses photons in the energy range 20 keV to 100 keV, a radiation band hereafter referred to as inspection band radiation or IBR. [10] The inspection radiation band is a sub-band of the X-ray spectrum which ranges from 0.1 keV to 100 keV. A typical setup has an emitter and detector somewhat analogous to a light and a camera. The detector, unlike a camera however, generally has a quite large, flat detection surface roughly equivalent in size to the objects being tested. A sample to be tested is placed on or near the detector such that it is between the detector and emitter. Radiation from the emitter therefore must pass through the target sample to reach the detector. Some of these radiant photon's will pass while others will not. The rate of flux attenuation along a path from emitter to detector is dependent on the material properties along that path, allowing the inspection of internal structure of the sample. After detection, the obtained measurement takes the form of a 2-dimensional gray-scale image. This image maintains the spatial relationships of the test material in the two dimensions of the flat detector, but projects spatial relationships in the test material in the third dimension onto it's 2-dimensional plane. This results in almost complete loss of spatial layout in the 3rd dimension.

Currently radiation tomography is used in food production lines to scan for foreign materials such as metal or glass. This exposure occurs after a batch of product has

been completely processed and bagged for shipment, and is therefore an ideal use case for non-destructive testing as physical inspection is no longer possible. It also ensures that the contents under exposure are not able to be removed nor new contents added as the bag is sealed before exposure.

For fried potato product, a service contract is maintained with each of the customers defining the distribution of lengths of product that each bag must contain. Product are binned according to length and are released to the bagging process according to the correct distribution. However, stochastic mixing and breakage during the bagging process causes some bags to not meet the specification. In order to determine the length distribution of the final product, the current procedure is to periodically inspect a bag's contents in full. This involves pulling the bag from the production line, opening it and measuring the length of each cut. The measurements from the inspected bag are then used to release or hold the 40,000 lbs of product surrounding it. Using a single randomly selected bag as a proxy for nearly 6,700 other bags likely leads to significant errors in both holding good product and releasing product that does not meet the required specifications. In addition, it does not reveal the detailed and timely information about the product lengths that would be helpful in guiding changes at earlier points in the production process. Currently changes to final product caused by changes made early in the production line can only be approximated by the sampling used, and only then after a time lag as the sampled bag is manually inspected. Because the foreign material tomography is performed after all processing is completed and product cannot intermix once the bags are sealed, it provides a good opportunity to measure the true distribution of lengths per bag. It could also provide information about the lengths that is immediate and specific to each bag.

Potatoes have an IBR attenuation density very similar to water [9]. A cooking oil, such as peanut oil for example, likewise has an IBR attenuation near that of water [11]. This makes their IBR attenuation much different than that of the air [11] that fills the rest of the bag. Because of this fact, the edge of each cut, and therefore the interface between potato/oil and air should present a differential in transmitted flux to the detector and show as an edge in the resulting image. Because radiation tomography preserves spatial relationships in 2 dimensions it is a good candidate method for determining length measurements. In addition the dimensions of a filled bag is such that one dimension is significantly smaller than the other two. The bags are exposed in an orientation where the smallest bag dimension is the dimension whose spatial relations are lost by the radiation tomography process. Because of this fact it is expected that there will be a minimum of lost length information.

### 1.3 Deep Convolutional Nets

Feed Forward Neural networks are a long studied approach to the problem of modeling using machine learning. They consist of a directed acyclic graph of neurons connected with weighted edges. Each neuron sums the values on its incoming connections, computes an activation function on this sum and emits this activation onto the outgoing edges. The weights of the network are then trained using one of many variations of a gradient descent type of training. This type of training uses the fact that all functions in the network are differentiable to adjust weights in the graph in a way that lowers the error of the final output.

A neural network with a depth of 2 can provably approximate any function with arbitrary precision given an unbounded number of neurons in the middle layer

[12]. Recent advances in training of deeper networks have allowed for unprecedented performance. In particular, convolutional neural networks (CNN) with many hidden layers have come to outperform most other techniques in tasks involving images. These include image classification, object detection and segmentation. Though less theoretical work exists describing these networks as for other techniques, they are regarded as networks that are, in some cases, capable of learning a set of hierarchical features, and thus relieve the practitioner of the burden of hand crafting features for the problem domain.[13] [16] These networks employ a set of small filters that are convolved over the image. This has the effect of tying the weights in the connection graph wherever a filter is applied. This technique vastly reduces the number of parameters in the model and makes the learned features shift invariant.[15] Because of the dominant performance of CNN's in the image domain, this approach is used herein to classify the content of the tomography images.

## 1.4 Thesis Statement

A single 2 dimensional radiation tomography exposure of an aggregate of fried and bagged batonnet cut potato has enough information content to recover estimates of statistics regarding the lengths of the aggregate it was generated from. Furthermore, these estimates can be obtained in general for exposures generated from the same process and different batches of aggregate using a model learned via annotated examples.

## CHAPTER 2

# RELATED WORK

Lotfi et al. use a neural network on microscopic images to classify potato chip quality.[18] The images they use are visible spectrum images and are processed at a microscopic scale for color and texture to yield a quality measure. Despite sharing the general type of prediction model chosen and the subject of interest being potato related, there is not a great deal of crossover between their work and this work. The most limiting factor here is simply the scale of the features they are using. While they are looking at microscopic images, this work is using images including as many as 1000 whole cuts. They train their very modest network on only 5 abstracted input features taken from the image histograms. Here on the other hand the entire image is being used, 193,800 pixels each taken as a feature along with several additional feature layers.

Mellema has some work indicating that fried potato product are composed of around 14% fat. This suggests that the product IBR attenuation might be more influenced by the starch and water components and less so by the oil it has absorbed.[19]

Miri et al propose the use of microscopic X-ray CT reconstruction to inspect the microscopic structure of fried potatoes to deduce porosity structure, crust size and uniformity to inform cooking temperature and processes.[21] They use these X-ray images to derive physical distance measurements similar to the goal of this work. There are three critical points that make their work not transferable to our problem. Primarily they work with images taken at a microscopic scale and as well make the distance measurements by hand rather than attempt to automate the process using a trained computer model. Although their images contain overlapping

porosity structure, they reconstruct a cross-section free from overlap using a "specially developed algorithm" for which no details are given.

Mohsenin has some results using radiation tomography to separate potatoes from rocks and dirt clods. Although few details are included, those results indicate that the X-ray Linear Attenuation Coefficient (LAC) for potato is substantially different than that of air and leads to a similarly different amount of flux across a variety of different energies.[22] They also suggest that potato X-ray LAC is substantially different than both dirt and rock.

Though the LAC for the rock referred to in Mohsenin's work is not necessarily known, work by [29] allows one to give a good estimate. They compute the LAC range for a sample of garnet schist to range from about 2 to 14  $cm^{-1}$  for the various components. They also estimate the LAC for quartz and silica to be 3.6 and 3.0  $cm^{-1}$  respectively under a radiation beam of 25 keV. More details about the implications of these measures are in section 3.6.1. In short they allow one to put a bound on the amount of IBR flux expected to pass through a bag at inspection time.

Shahin et al use edge and Discrete Cosine Transform features on radiation tomography of onions as an input to a neural network to classify internal defects.[25] Mery et al use Gabor filters and neighbor pixel co-occurrence matrices fed into a neural network to detect defects in welds from radiation tomography of those welds.[20] In both cases the object of the study differs quite a bit from the product studied here. However, both cases share an important similarity. Their results are derived from a combination of static features that focus on edge and texture qualities of the images. Also in both of these works the features are then fed to a neural network to learn a prediction model. We choose here to integrate their strategies in using edge and texture features fed to a neural network.

# CHAPTER 3

## DATA GENERATION

### 3.1 Artificial Image Generation

Because of the large number of parameters in a deep net and the high dimensionality of the inputs, a large amount of annotated data is needed for training. The process of physically annotating images from the production line is a costly one in terms of time and manpower, making it infeasible to collect all the needed training data from the plant. For this reason choosing to generate artificial images to serve as training data presents several benefits. Because the images are generated in simulation, any relevant detail about them can be relatively easily recorded and used during later processing. This fact holds in the case where a detail is desired well beyond the time that the images were created. For real images from the plant, once data has been collected and the product disposed of, the opportunity to collect additional pieces of information about them is lost.

Annotated artificial images can as well be generated much faster than can the annotated real images. Estimates for the time to annotate a real image are around 20 minutes in the plant which involves a human worker and specialized equipment. An annotated artificial image takes in the neighborhood of 7 minutes to produce on a single CPU core. Although this base time is not much faster, it is much easier to scale the production up with more resources. Annotating more real images would require more workers and access to more of the specialized hardware needed to measure the lengths. To scale artificially requires access to more CPU cores. Already, most modern computers have anywhere from 8 to 32 processing cores, each of which can

produce an image in the roughly 7 minutes. Creating more images only necessitates acquiring additional commodity computers.

Training with artificial data also allows more control over the training dynamics. For most machine learning systems an imbalance in the types of training examples presented can cause poor final performance on the minority data types. For a production operation like Simplot, many efforts are made to minimize the time that out of specification goods are produced. The result is that the vast majority of product that goes through the tomography machine will be at or very near the target product specification. Any bags that are far from that specification are not likely to be chosen at random for annotation and even in the event that they are will be hugely outnumbered by the near to specification product. The expectation for a system trained on data obtained this way will be that it performs poorly on out of specification product images. This is hardly a desirable trait. With the capability to generate artificial data, comes the capability to balance the types of data presented to the learning algorithm and maximize the chances of successful operation on both in specification and out of specification product.

Lastly, from a more academic standpoint training with artificial data provides an opportunity to investigate the caveats of training a learning algorithm on artificial data and testing it on real data. Not many domains have this opportunity. In most domains artificial data would not be realistic enough to serve its purpose. This is certainly true for natural language, natural images, speech, and likely for social network data to name a few examples. Additionally it is often prohibitively computationally or time expensive to produce such data. Here there is the rare opportunity to be able to produce accurate and relatively fast artificial data.

The last two points highlight the challenges here. What examples should be

generated and how realistic do they need to be to serve their purpose knowing that the price paid for that realism is time? The following sections provide the details but in short, two overarching data-sets were produced. A historical data-set was designed to represent the data that would be most common in the plant and is close to the target specification. Complimenting that is a non-historical data set that is conceptualized as a random sampling of "everything else". The general strategy for realism was to start simple and iteratively make adjustments to bring the images more in line with real images as guided by the CNN performance and image statistics of the two groups.

## **3.2 Distribution Modeling**

Before any artificial exposures are generated, a model for each product detail is made. Each bag has nearly one-thousand cuts in it. Several distributions are detailed from which those cut details could be drawn. Those distributions provide the width, density, and length of each cut created. Each of these distributions is based on a list of 4351 samples taken at the plant. This list is assumed to be an unbiased sampling of the bags seen during normal production.

### **3.2.1 Width Distribution Modeling**

A width distribution is used based on feedback from measurements at the plant. The claim is that the widths vary from 0.25" to 0.29". This information could have been modeled in many ways as it is somewhat ambiguous, but any model used would need to enforce those bounds in some way. Liberty was taken in assuming that those measures are not hard boundaries but rather measures that describe the majority of

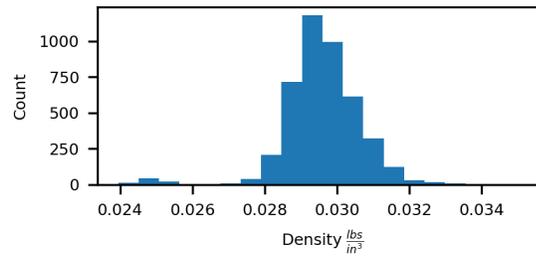
the product. With this assumption the cut width for the generative model is sampled from a gaussian distribution with mean 0.27" and standard deviation of 0.01". This model provides that as large amounts of product are made, very close to 95% will have a width between the measures cited by the plant. It should be noted that this model is used to sample two of each cut's three dimensions as each cut has a nearly square cross section.

### 3.2.2 Density Distribution Modeling

The historical data shows that with a very small margin of error, each bag weighs 6 lbs. Therefore, in order to properly generate artificial bags each cut's weight would need to be computed. This necessitates having a model of density. Again the historical data is the source of this model. The historical data details the mean weight per inch of product length in each bag. With the assumption from above that the mean cut width is 0.27" one can compute the mean density which is then simply the ratio of the weight per inch of length to the mean cross sectional area.

$$\bar{\rho}_i = \frac{\bar{\omega}_i}{0.27^2}$$

where  $\bar{\rho}$  is the mean density,  $\bar{\omega}$  is the mean weight per inch length. Visual inspection of a histogram of these estimated densities shows for the most part a nearly gaussian distribution as can be seen in Figure 3.2.2. Analysis of these densities yields a mean of  $0.0296 \frac{lbs}{in^3}$  and a standard deviation of 0.00108. Although it is unknown if this is an accurate distribution for the density the product, it is used as a proxy. These values parameterize a Gaussian distribution modeling the density which is sampled for each cut generated. Because the utility of such a measure is in estimating how many cuts



**Figure 3.1: A histogram of historical product density.**

will need to be sampled to reach a full 6 lb bag, it is not necessary that the estimate be accurate per cut, only that it be accurate in aggregate. Because each bag has nearly 1000 cuts within, one can be assured that the final result will be very nearly correct via the law of large numbers.

### 3.2.3 Length Distribution Modeling

The distribution of lengths in a bag is a slightly more complex topic. This is due in part because of the nature of the historical distribution itself and in part because it is the focus of the predictive model and deserves the additional attention. The historical data is presented in the form of product counts in 5 mutually exclusive logical bins. Each bin details the lengths of the product that contributed to its count. It is perhaps not initially clear that the true dimensionality of this distribution is 4 instead of 5. However, because the bag weight is fixed at 6 lbs there is a constraint that the sum of the 5 bins be the correct value to achieve the 6 lb target. This constraint brings the degree of freedom and effective dimensionality to 4. Rather than analytically project the historical 5 bins onto a 4 dimensional space, the historical data was modeled using principle component analysis (PCA). This revealed that indeed the length distribution could be very well represented in a 3 dimensional

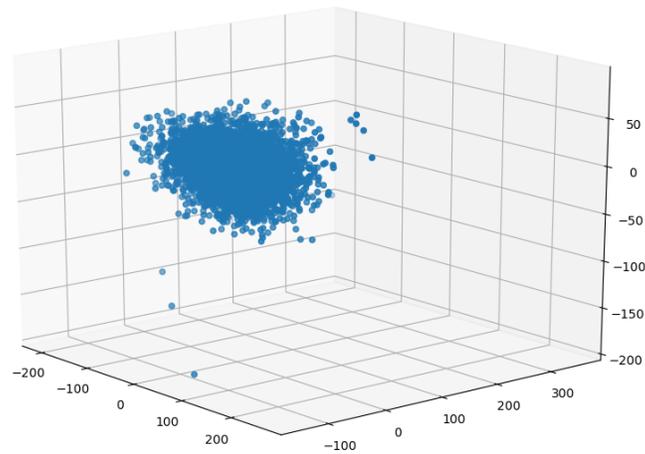
space retaining 91% of the original variance. It is fortunate that 3 dimensions is so adequate because it lends itself to visualization of the data. The left side of Figure 3.2.3 shows the resulting 3 dimensional representation. It is quickly apparent that there is a single cluster representing the vast majority of the data and a handful of outliers. Visual investigation of the cluster reveals structure that is very conducive to multidimensional Gaussian modeling, in this case a Gaussian mixture model (GMM). This is represented by a generally spherical or oval shape. It is additionally desirable to model this data with a GMM because the intent is to generate samples and a GMM is easily sampled from. Probing deeper and looking at the next PCA component reveals a small partition in the data into one large dominant cluster and another small cluster. This can be seen in the right side of Figure 3.2.3.

Ultimately the data was reduced to 4 dimensions because this strategy retained nearly 97% of the variance and preserves the small separated cluster. A Gaussian Mixture Model is fitted to the 4 dimensional PCA distribution using the Bayesian Information Criterion (BIC) for parameter selection. The BIC attempts to characterize the likelihood of the data under the model in comparison to the model complexity. It decreases with quality of model fit and increases with the number of free model parameters. In particular it takes the form:

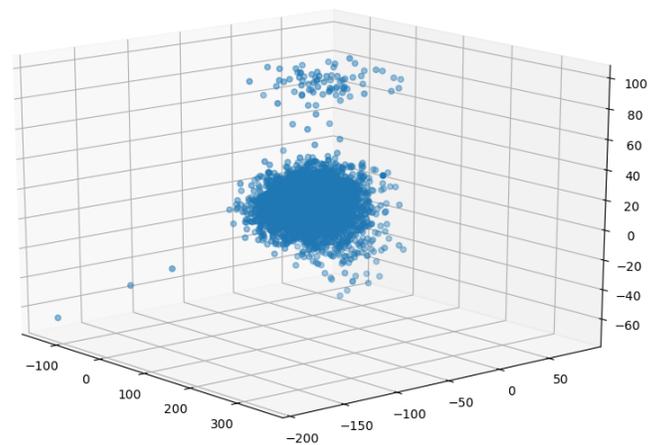
$$BIC = -2 \cdot \ln(\mathcal{L}(D|M)) + k \cdot \ln(\|D\|)$$

where  $D$  is the data,  $M$  is the model,  $\mathcal{L}$  is the likelihood, and  $k$  is the number of free parameters in the model. Using this criteria helps generate a model that doesn't simply overfit the given points but generalizes their structure. This is helpful since the plan is to draw samples that are characteristic of the historical data but not the same

Historical Observed Fry Length Distributions Top 3 PCA Components (91% variance)



Historical Observed Fry Length Distributions PCA Components 2-4 (97% variance)



**Figure 3.2:** Projections of the 5 bin historical data onto the first 3 of 4 PCA components and the last 3 of 4 PCA components respectively. Note that the axes are each some linear combination of the original 5 bins and thus are not easily interpretable.

**Table 3.1: Weights of the 5 Gaussian mixture model components used to model the bag length distributions**

Component 1	Component 2	Component 3	Component 4	Component 5
0.14337062	0.01857595	0.22006299	0.31216054	0.3058299

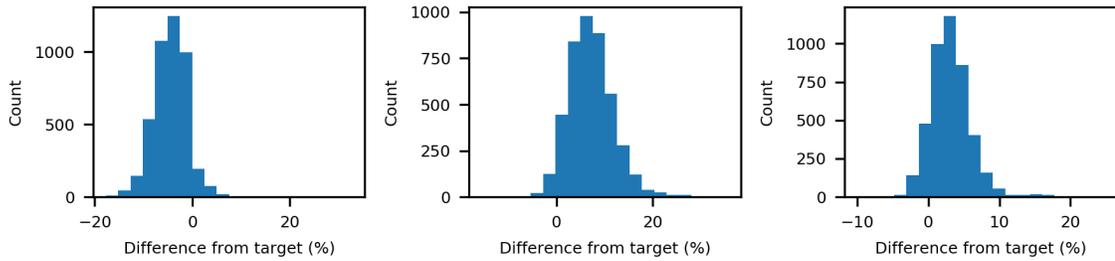
**Table 3.2: Mean bin counts for each of the 5 Gaussian mixture model components.**

GMM Component	> 4"	< 4" & > 3"	< 3" & > 2"	< 2" & > 1"	< 1"
Component 1	28.1	200.3	393.4	248.0	118.4
Component 2	24.8	122.5	325.9	302.5	229.0
Component 3	25.8	193.3	411.2	247.6	120.0
Component 4	22.3	165.6	371.1	267.2	128.0
Component 5	24.4	174.3	356.6	247.7	133.0

as them. Although the 3 dimensional data appeared to have 2 dominant clusters the BIC criteria and the 4 dimensional reduction yielded 5 Gaussian components. The relative contributions of each of the components is shown in Table 3.2.3. The center or mean of each of these components can be seen in 3.2.3 One critical observation here is that component 2 plays a very small role in the overall model, 1.9% to be exact, from the table of weights. Additionally, inspection of the means shows that it is quite a bit different than the other 4 components. It is assumed that this component is representing the few outliers that exist in the data.

### 3.3 Distribution Sampling

As mentioned above, these distributions alone are a very narrow view of what is possible in the plant since they reflect what is most common. The historical list confirms that most bags are at or near the specification and highlights the need for a non-historical distribution. Figure 3.3 shows this relationship. Each of the three



**Figure 3.3: Histograms of the distribution of the historical data used for modeling relative to 3 aspects of the target specification.**

histograms shows the distribution of the historical data around one aspect of the target specification. Each of these aspects is measured as a percentage of the total bag. As can be seen, the majority of samples are within 10% of any aspect of the specification, and that is only on one side. This fact is as well reflected in Figure 3.2.3 where all but 5 examples are a member of a dense cluster. For this reason two sampling strategies are adopted, one to acquire historically likely examples, and another for historically unlikely ones. In either case the sampling of width and density are the same. Only the length sampling changes.

### 3.3.1 Historically Likely Samples

In order to acquire historically likely samples, the GMM model is sampled directly to gain a 4 dimensional vector. This sampling is a two step process. Initially a GMM component is selected at random using the component weights as bias. Then the individual multivariate Gaussian distribution is used to extract an individual sample in 4-d space. This involves sampling from a 1-d Gaussian 4 times to create a 4 element vector. This vector is then multiplied by the 4 orthogonal eigen vectors that define the component scaled by their respective eigenvalues. After adding the component means, a 4-d sample is yielded. This 4-d sample vector is then multiplied by the

transposed PCA transform matrix to result in a 5 dimensional bin count vector in the same form as the original historical data.

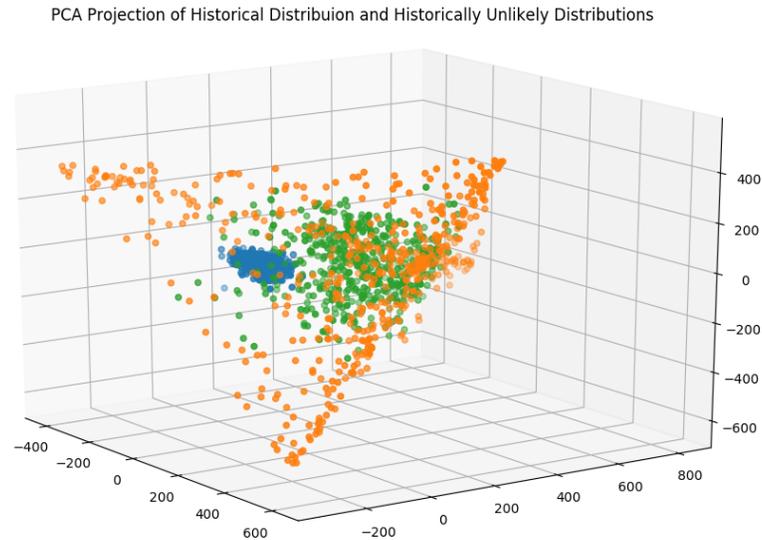
### 3.3.2 Historically Unlikely Samples

Obtaining historically unlikely length distributions is a more complex process. Ignoring infinitesimally small probabilities, the historical GMM describes a probability over a set of bounded, analytically defined spaces from which one can sample. The opposite of this, historically unlikely samples, consists of a probability distribution over all possible space that is the inverse of the GMM. Certain aspects of this distribution are easily defined.  $P(x)$  for any  $x < 0$  must be 0 since negative length cuts are impossible. The upper bound is however not so easy since it depends on the sum of the product weights to reach 6 lbs. Worse than this is the fact that the distribution be inversely related to the sum of the 5 GMM components. Overall analytically defining such a distribution would be very difficult, much less properly sampling from it. Considering the purpose of this task which is to simply generate bags that are unlikely to occur in regular production, the effort required to create such a distribution is not warranted.

The compromise then is perform a sort of Monte-Carlo search for unlikely bags using the GMM as a rejection criteria. This process proceeds by selecting a set of bin counts at random from a flat distribution. This itself is a rather difficult problem as the 5 counts are constrained that their sum come to 6 lbs. Historically this means that the total count is roughly normally distributed with a mean of 982 and standard deviation of 52. Creating a sampling technique in which any combination of bin counts that satisfies this constraint is equally likely is not feasible so two different strategies are employed, each with a different compromise. The first, a scaling strategy, is to generate 5 i.i.d random numbers from a flat distribution in the range 0 to 982. These

values would then be scaled such that their sum is 982. This scaling has the result that the 5 final bin counts are no longer i.i.d and favors central values with  $982/5$  being the most likely. The second, an iterative strategy, is to sample each value in turn from a flat distribution in the range  $[0, \text{remaining}]$  where remaining is 982 minus the sum of already established values. This allows complete freedom in the first value sampled, followed by a tendency toward smaller values as more of the 5 vector is filled in. Because this process has a bias in terms of the order of values sampled, 5 values are generated in this manner, then randomly shuffled to create a resulting 5 vector. Between these two strategies the sets of centrally tended vectors and vectors skewed toward one bin are represented. Regardless of the strategy used, the resulting 5 vector is then transformed via the PCA model and the resulting 4 vector scored via the GMM for likelihood. The bin count is accepted if it is twice as likely under a flat distribution as it is under the GMM. For this purpose a flat distribution was defined as  $\frac{1}{982^5}$

This strategy is an admittedly crude approximation from an analytical probability standpoint. However, the intent here is not to accurately represent the probability space, but rather to generate a broad spectrum of distributions in order to increase the generalization capability of the final regression model. Figure 3.3.2 shows a PCA projection into 3d space of a sampling of the distributions generated. The Historically unlikely samples span a space that both covers the historically likely samples yes is much broader as intended. It is also important that this is achievable in a reasonable amount of time. In practice this technique is quite fast. As mentioned before the historically likely data represents a pretty narrow subset of what is possible. Therefore sampling from a flat distribution is unlikely to result in a bin distribution that is likely under the GMM. In the event that it does, a resample usually fixes the problem.



**Figure 3.4:** Comparison among the distributions of historically likely and both strategies of historically unlikely distributions. The points plotted are actual example samples. Historically likely samples are blue, scaled non-historical samples are green, and iterative non-historical samples are orange.

Anecdotally, most samples were accepted upon first draw, with a second sample being needed not very often. Four samples is the most seen anecdotally.

### 3.3.3 Sampling Lengths from Bin Level Distribution

The native format of the Historical length distributions, a 5 value vector of bin counts, is not sufficient for generating a set of actual cut lengths. The primary problem with this format is that there is no specification of the distribution of the lengths within a bin. To solve this a two stage process is implemented to sample actual cut lengths

similar to the process used to sample from the GMM. The first step is to normalize the bin counts to be a proper probability density function such that they sum to one. This is done simply by dividing each by the overall sum. This probability density function (PDF) is used to make a cumulative density function (CDF) by specifying the lower CDF value for each bin to be the sum of all the pdf values for the bins before it. This way a single random value from a flat distribution on  $[0, 1]$  can be mapped through the CDF function to a bin. This allows one to choose a bin to sample from consistent with the overlying distribution.

Because only the minimum and maximum length of a bin is known, to sample within the selected bin some assumptions had to be made. The primary assumption here is that the probability assigned to a bin is for the central length value for the bin. As an example, if there is a 0.3 probability of sampling a length in the 3" to 4" bin, that results in a value of 3.5" mapping to 0.3 in the final continuous pdf. This way one can perform linear interpolation between these values to obtain a continuous pdf. Linear interpolation has two valuable outcomes. One is that it expresses the intuition that the probability at the extent of one bin should be the same as the probability in the adjacent extent of a neighboring bin. As an example if the 2" - 3" bin has a 0.2 probability and the 3" - 4" bin has a 0.8 probability one should expect a 2.99" cut to be nearly the same likelihood as a 3.01" cut. This likelihood should as well be somewhere in between 0.2 and 0.8. The second main benefit of linear interpolation is that it results in a piece-wise linear pdf over the bins which is easily used to create a CDF and thereby sample from. The creation of that piece-wise linear pdf requires one additional assumption, specifically that 0" cuts and cuts over a certain length are impossible. In the historical data, the average bag contained 0.8% product longer than 6.5" and 13% product longer than 4". With a linear interpolation, this implies

a maximum product length of 7.3". To make the bin lengths easier to interpret and to allow for a little more outlier lengths, it was assumed that the probability of a 0" cut and an 8" cut were both 0 when performing the linear interpolation. Linear interpolation is performed as follows for a particular selected bin. The two neighbor values  $n_0, n_1$  are used as vertices of a parallelogram to represent the bin. The Area of this parallelogram is found and both values are scaled such that the parallelogram area is a proper pdf summing to 1.

The upper edge of this parallelogram is used as a linear pdf with equation.

$$P(l) = \frac{n_1 - n_0}{w}l + n_0$$

where  $w$  is the width of the bin. To sample from this a CDF must be used. This CDF is the integral of the linear pdf.

$$CDF(l) = \int_0^w P(l)$$

For a linear pdf the CDF is quadratic and a sample can be found by generating a random value from a flat distribution over  $[0, 1]$  and using the CDF inverse.

$$l = \frac{-n_0 + \sqrt{n_0^2 + 2mn}}{m}$$

where  $n$  is the random value and  $m = \frac{n_1 - n_0}{w}$  from the linear pdf.

With length, width, and density sampling capabilities, individual cut details can be generated in mass. For each bag, cuts are generated one by one and for each a weight is computed by

$$weight = \rho \cdot l \cdot w^2$$

Once the weight has surpassed the 6 lb target, generation for that bag ceases, a unique series id number is given to it, and the details are written to disk for later processing.

### 3.4 Mesh Generation

The next stage in the process involves reading the details of a bag's contents and generating a virtual model for each cut. These models are in the form of a set of vertices and faces called a mesh. Specifically each vertex is specified as a 3 tuple coordinate in 3-d space. Each face is specified as a list of vertices which it spans. For each cut the Blender software package [2] is used to create a cube spanning  $[-1, 1]$  on each axis. The cube is then scaled in the z and y axes to match the width of the cut. The cube is then scaled in the x direction to match the length of the cut. Once the basic dimensions are achieved the mesh is partitioned into 6 segments lengthwise resulting in 28 vertices and 26 faces. For some meshes, this is all that occurs and they are written to disk.

Stochastically, 50% of the cuts undergo a curvature operation. This operation generates a spline with 5 points. The center point is at the center of mass of the mesh. The other points are distributed evenly in both directions along the length of the cut with offsets in the y-z plane from a Gaussian distribution with standard deviation of 0.4". This spline is then used as a curve around which the mesh vertices are translated. The end result is a mesh that follows the the path of the spline. The amount of true curvature in the product coming from the plant is unknown. The measures used here are an estimate based on visual inspection of tomography images and an actual photograph of a bag. these meshes are given a mesh id number and written to disk.

### 3.5 Simulation

The meshes must be placed in an orientation that would naturally occur during bagging in order to generate a realistic exposure image. The Bullet [4] physics simulation software package is used to simulate the falling of product into a container. The system chooses one of several predefined containers matching the size and shapes the plant bags end up in. The set of meshes for a given series id are read from disk and stochastically placed in a column above the container in a range from 5" to 40" vertically and in a Gaussian disk with standard deviation of 2" horizontally.

The simulation is done in steps. During a step, the velocities of a mesh are used to compute it's location for the next time step. Using large time steps provides rapid simulation but is inaccurate and results in tunneling. Tunneling occurs when imperfect collision detection occasionally resolves a collision by tunneling a mesh through another. This periodically results in a mesh tunneling through the container only to then fall below it endlessly. This effect can be avoided by using many very small time steps for simulation. The tradeoff is that the simulation can take a very long time. A balance between these two factors is made by lowering the time step granularity enough to allow some degree of tunneling. Tunneled meshes below the container bottom are detected via a simple scan of the center of mass position every simulation step. These meshes are placed just above the height of the highest mesh and simulation continues.

Inspection of real images from a production line reveals that many of the bags end up in non-straight orientations at inspection time. They can be rotated among any of 3 axes: pitch, roll, and yaw. To produce the yaw effect in the artificial images, The container is rotated randomly using a Gaussian sampled value before simulation

begins. A guide exists in simulation to help reduce the number of meshes and fall below the container making rotation in pitch and roll not as simple. Thus, pitch and roll effects are produced by rotating the gravitational vector by a normally distributed random amount in the x and y axes. (z being vertical)

A simulation is run which involves a gravitational pull toward the container and the detection and resolution of collisions among meshes and container. The simulation ends when the maximum linear velocity and angular velocity of the meshes falls below a threshold. This is an indicator that the meshes have mostly settled. Early termination in this way results in much faster simulation while still resulting in a realistic end orientation.

This process is repeated 33 times for each set of meshes to simulate a series of bags from a stationary distribution. Simulating the same meshes repeatedly ensures that there is substantial random variation in the output images, but that the distribution of lengths in the images will be exactly the same. This provides the correct guarantees later for statistical analysis under the assumption of a stationary distribution. Once completed, the resulting meshes are written to disk to await simulated Exposure. An example of such a mesh can be seen in Figure 3.5

### **3.6 Exposure**

The simulated mesh orientations represent the final resting place of bagged product and must undergo simulated radiation exposure to produce a final image.



**Figure 3.5:** An example of the final orientation of a set of meshes after simulation.

### 3.6.1 Radiation Attenuation

In order to simulate an radiation tomography, the Hounsfield unit value for each object must be known. The Hounsfield unit is a measure of Radiation attenuation for a particular substance derived from the natural linear attenuation coefficient. It is a common measure used in medical applications and is derived in such a way that the measure for pure water is 0 and that of air is -1000. The formulation of this derivation is seen in

$$HU = 1000 \cdot \frac{\mu - \mu_{water}}{\mu_{water}}$$

Where  $\mu$  is the linear attenuation coefficient at a given radiation energy. The linear attenuation coefficient is defined as the probability that an individual photon will undergo an interaction with a material during travel through a given length of that material. Its units are  $cm^{-1}$ . This value is dependent not only on the material but also on the photon energy. Table 3.6.1 shows some values for elements and compounds relevant to this work.

Brown et al.[3] found the linear attenuation coefficient of vegetable oil to be  $0.142 \text{ cm}^{-1}$  at 75-80 keV and  $0.132 \text{ cm}^{-1}$  at 140 keV placing the Hounsfield value for vegetable oil at -106 to -114. This is valuable because the product in question here is estimated to be about 14% oil. Other than water, whose Hounsfield value is known to be 0, the only remaining component for which a Hounsfield value is not known is the starch. A study by Cohen et al. [6] illustrates that an iodized starch suspension in liver increases the liver's naturally low (40-60) Hounsfield value by about 180 to 200. The computed Hounsfield value for iodine is extremely high, suggesting that the increase isn't necessarily attributable to the starch. The fact that the linear attenuation coefficient is a function of the density of the material also suggests that indeed the Iodine with its high atomic mass, is the cause of the effect in liver and not the starch. Though somewhat uncertain, it is assumed that the starch component contributes a minor positive value to the overall Hounsfield value.

Aluminum is used as a calibration material in determining some of the unknown values in the artificial image pipeline since some of the real images have a known aluminum sphere present in them. The NIST physical measurement laboratory mass attenuation coefficients database places Aluminum's  $LAC(\mu)$  at 0.5446582 at 80 keV and 0.3719222 at 150 keV resulting in a Hounsfield of between 2425 and 1496. [28]

Though the Hounsfield value for fried potato is not present in available literature nor easily calculated from available information, it is ultimately a mixture of water, oil and starch. The vast majority of a potato's density before frying is water. During the frying process, some of that water is lost and replaced by oil. This makes the Hounsfield a weighted mixture of -106, 0 and some likely low positive value for an 80 keV beam. For the purposes of training a network to recognize lengths, it is not critical that this be exact. We simply need the resulting image to be correct. An error

**Table 3.3: Linear attenuation coefficients  $\mu$  and Hounsfield unit values (HU) for relevant substances.**

Substance	$\mu$ @ 80 keV	$\mu$ @ 140 keV	HU @ 80 keV	HU @ 140 keV
Vegetable Oil	0.142	0.132	-106.9	-114.1
Aluminum	0.5446582	0.3719222	2425.52	1496.122
Iodine	17.15	3.381	106861.63	21691.2
Water	0.159	0.149	0	0

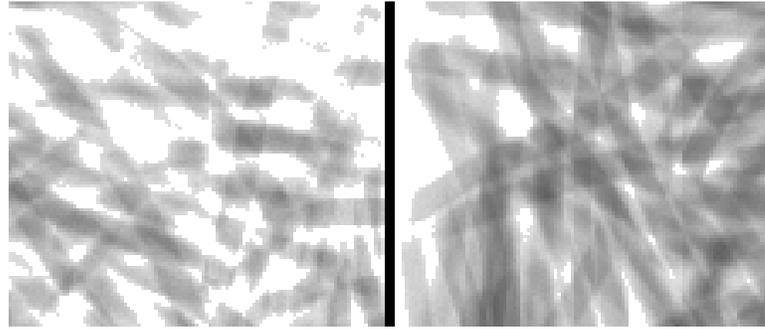
in this value can be corrected by an adjustment in the detector sensitivity or radiant beam energy. Here it is assumed that the product in question has a Hounsfield of -50. Image based validation described later performs the feedback necessary to correct the images in the event that this is incorrect.

### 3.6.2 Detection Sensitivity

Several observations drive the parameterization of the detector. The real output images show white, a grey-scale value of 255, in areas where no occlusion is present. The artificial detector simply collects a flux value without bound. A value is set to determine the flux threshold at which a white value will be output. In this case any flux over the threshold of 0.0767 outputs a 255 to the resulting 8 bit grey-scale image.

Unusually, although the real images show 255 for un-occluded regions there is a gap between that and the next value output of 224. For reasons unknown that band of values does not occur in real images and must be replicated. All flux values not over the threshold of 0.0767 are scaled linearly by a factor of  $0.878 * 255$ . This results in a mapping between flux and 8 bit grey-scale values where the maximum output of occluded areas is the maximum known to occur in real images (224).

An interesting effect seen in the real images is that of variation in the amount of flux reaching the detector under similar circumstances. An example of this is the

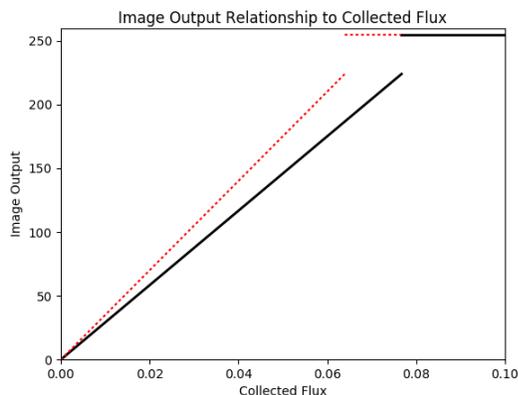


**Figure 3.6:** An example of the overexposure effect on the left compared with a normal example on the left.

circumstance where a single cut is occluding the detector. In certain images this yields a clearly identifiable gray region in the shape of the cut. In others a single cut only appears where other cuts cross it giving a washed out appearance similar to an overexposure in photography. Figure 3.6.2 shows such an effect.

Modeling the overexposure effect is accomplished by linearly scaling all flux values by  $1 +$  the absolute value of a Gaussian random value with mean of 0 and standard deviation of 0.1. Figure 3.6.2 shows the overall relationship between flux collected at the detector and the final output value.

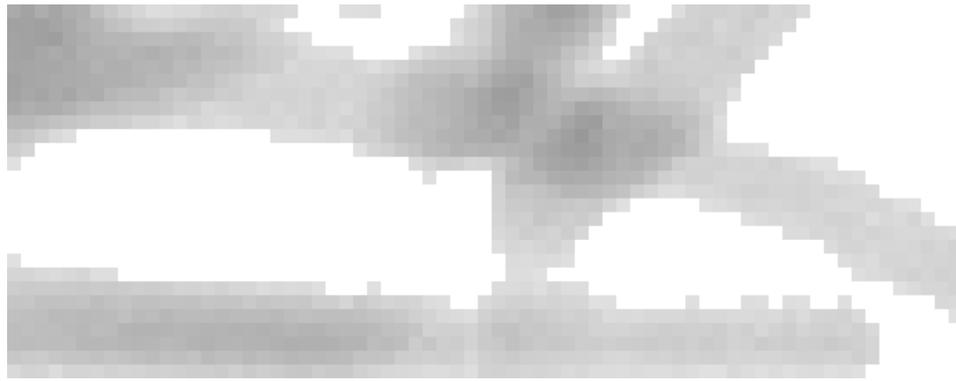
Haff et al. suggest that agriculture products are inspected using radiation energies in the range 20 keV to 100 keV, the IBR.[10] A set of real images are used containing a known 4.5 mm aluminum sphere occluding the detector to set this value in the virtual exposure. The output values in 10 real images in a 3x3 patch at the center of the sphere were measured and yielded the outputs 24, 27, 27, 32, 33, 33, 33, 38, 38, and 40. An artificial sphere of 4.5 mm was created and added to the top of the final oriented meshes. The IBR energy and sphere Hounsfield value were both adjusted until the output for the example was 33. The result was an energy of 80 keV and an aluminum Hounsfield of 4000. One will notice that these values are not in agreement



**Figure 3.7:** The relationship between collected flux and final image output value used in the artificial exposure process. The dotted line shows two standard deviations of randomness added to the relationship to capture the overexposure effect in the real images.

with those in section 3.6.1 obtained from the NIST data. Part of the difficulty here is that in each case, real and artificial, the aluminum sphere is above a column of product, so a measurement of it free of noise is not possible. The mismatch between the value used here and the ones contained in NIST is likely a compensation for the combined error in several of the other estimates used. Because there are so many degrees of freedom in parameterizing the artificial image pipeline where a raising of one value is compensated by a lowering of another there are likely many different combinations that will lead to acceptable images in the end. The fact that the value arrived at through image based validation is merely 60% larger than the NIST value suggests in a sense that other inferred parameters are similarly close to true values.

The output images show a consistent noise. This is most obvious in areas where only a single cut can be seen because it is not confounded by the presence of other cuts' edges. Figure 3.6.2 shows an example of this noise. As with most noisy signals the noise level is dependent on the overall signal level. The noise in these images is no



**Figure 3.8: An example of the noise present in the real images.**

exception. The darker regions, those with less overall flux collection, seem to exhibit less noise. To account for this the final collected flux values have a noise added to them that is scaled by the flux value. The final output is as follows.

$$output = f + (2r - 1) \cdot kf$$

where  $f$  is the flux at a given pixel,  $r$  is a random value on the interval  $[0, 1]$ , and  $k$  is a noise scaling constant. In practice a  $k$  of 0.03 gave artificial results similar to the real ones.

### 3.6.3 Spatial Parameters

Many of the spatial relationships in the example plant tomography machine are unknown. What is known by direct measurement is that the detector is 2" below the conveyor belt the bags traverse. Some measurements have been taken in an attempt to determine the emitter distance and size, but there is some uncertainty in the interpretation of those values. Regardless, much can be inferred through the images themselves. Much like an out of focus camera, a large or very near emitter can

cause a blurring effect. The real images show a little blurring in the product edges that would suggest either a moderately large or somewhat close emitter. In addition a very closely placed emitter would cause a magnification of parts of the image that are close to it. The plant images don't show any evidence of such magnification. This implies that the emitter is sufficiently distant to eliminate magnification and that the emitter has a non-negligible size. In simulation this is achieved by making the emitter 48 inches away from the base of the bag and square with 0.2 inches on a side. Because the detector is at 2 inches below the bag's base, the emitter is 50 inches from it.

Though not directly measured, the horizontal resolution of the detector has been inferred. This has been determined from inspection of the plant images with known product widths to be  $29 \frac{px}{inch}$ . This value is arrived at by dividing the pixel width of a cut in an image by the true cut width and averaging over several such samples. In the direction of motion of the conveyor belt the effective resolution is a function of the belt speed. The majority of the images show bag and product dimensions consistent with a  $29 \frac{px}{inch}$  resolution in this direction as well. However there are some images that are stretched in this direction in a way that is unlikely to be simply lengthening of the bag. This phenomena has not been investigated but could be due to slowing in the belt speed.

### 3.7 Image Based Validation

The production of artificial tomographic images involves many variables. The true plant value for many of these is not known, and some are simply approximations of some real physical phenomena. With this much uncertainty it is important to validate the process. Several statistics are taken over a set of images with the white borders

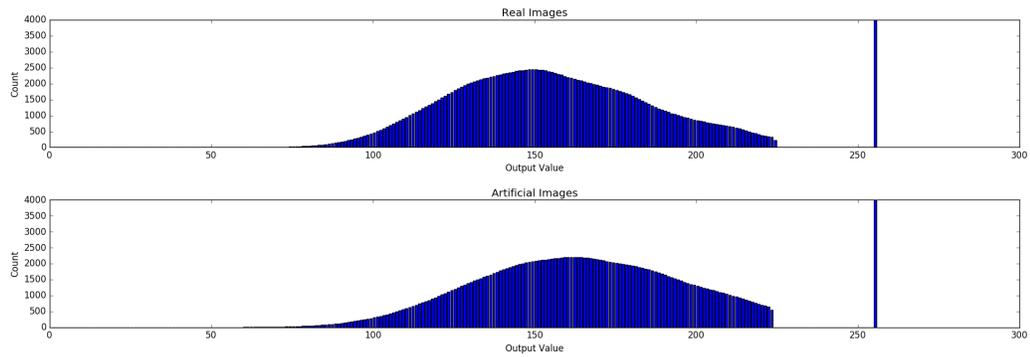
cropped out. The cropping process is described in section 3.8.1. These statistics are described below and are used to validate the processes that produced them.

### 3.7.1 Color Histograms

A color histogram is computed over the real images and over the artificial ones. This histogram shows the mean amounts of each gray level contained in the images. This is an important measure to show that one image set is not generally brighter or darker than another, and that the variety of colors represented matches closely. Figure 3.7.1 shows the results from a representative set of real and artificial images. There are two slight differences between the two sets. The artificial distribution is smoother than the artificial one which has a subtle peak just higher than the main peak. The second, and more substantial difference is that the artificial images display a small shift right in the histogram compared to the real ones both in the mean value and in the density in the right tail. Simply changing the collector sensitivity or the IBR energy is unlikely to solve the problem as it would simply shift the distribution left, thereby altering the left tail which matches the real images. What is more likely the culprit is too aggressive settings for pitch and roll in the simulator. Exceptionally high values for these settings cause the meshes to gather in one area of the container. The result is that one side or corner is left mostly bare causing an increase in white and near white pixel regions. All things considered, the artificial images match closely the real ones in their intensity distribution.

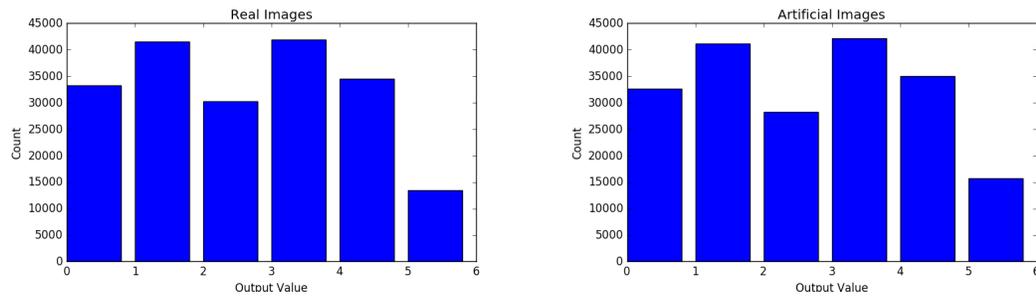
### 3.7.2 Linear Binary Pattern Histograms

Image intensity histograms, though useful, do not capture spatial information in the images. To validate that information Local Binary Patterns(LBP) are used [23].



**Figure 3.9: Tomographic image output value histograms for real and artificial images.**

LBP is a method to characterize the texture of an image. It labels each pixel by looking at the relationship between that pixel and some neighbor set of that pixel. The label assigned considers an arbitrary number of rotation invariant arrangements of neighbors at an arbitrary distance, and is determined by the relative intensity of the pixel compared to these neighbors. For this purpose a neighborhood consisting of 4 arrangements at a radius of 8 pixels is used. The radius of 8 corresponds to the pixel width of the average product width and is chosen so that the neighborhood of any given pixel is computed over the neighborhood around a mesh and not within it. The choice of 4 arrangements is chosen to match the parameterization used during feature extraction described in 4.1. For each set, real and artificial, the LBP features are extracted from each image and the mean count for each feature value is computed over the set. Figure 3.7.2 shows the results of these counts, and indicates that in terms of image texture, the artificial images match the real ones.



**Figure 3.10: Local binary patterns average histograms for a set of real and artificial images.**

### 3.7.3 Measures of Central Tendency

The final set of statistics taken are the per pixel mean, variance, and skewness over the cropped images from both the real and artificial sets. Figure 3.7.3 shows these results. These statistics give a good visual indicator of where differences and similarities might be spatially over the image sets. Because the artificial statistics have been taken over 1388 images and the real over just 211 the former will be less susceptible to individual variations causing the smoother textural appearance. Here, the indication is that the real images have a bias toward higher variation on the right edge of the image, resulting in a skewness on the right side toward lighter values. This bias does not show in the mean images indicating that it is a somewhat rare but extreme occurrence. As can be seen from the artificial images every edge has a high variance due to the uniform chances of pitch and roll used during simulation. It is unclear whether the bias in the real images is due to a physical layout phenomena like a bag roll bias or to a bias in the right side of the inspection machine to "overexpose". Much of the difference in the images could be closed by biasing the artificial roll toward the left and reducing the pitch probability. Regardless, the mean value spatially for these two sets is quite close.

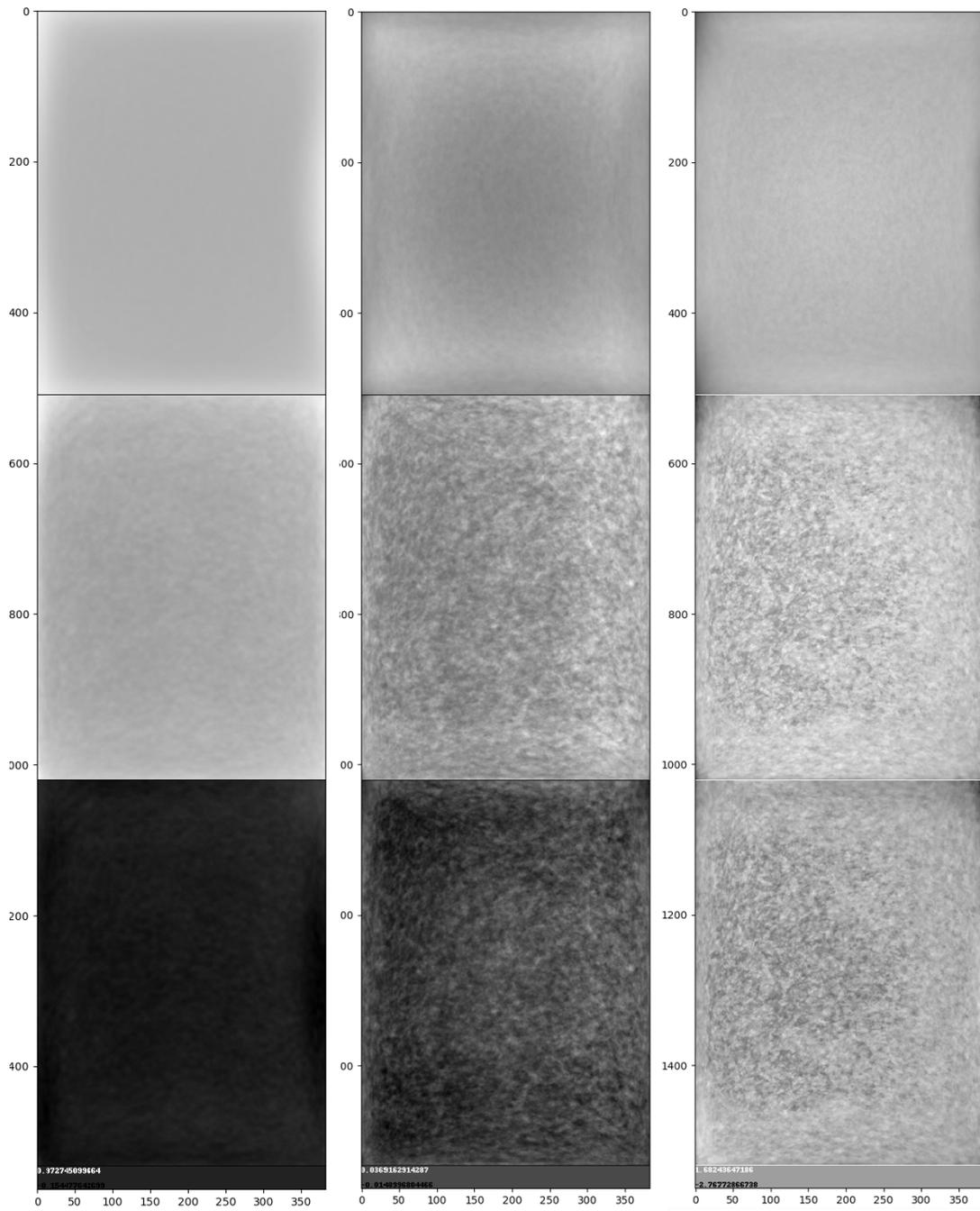


Figure 3.11: Per pixel image statistics. From left to right: mean, variance, skew. From top to bottom: artificial, real, difference

## 3.8 Image Staging

Many artificial images are created, around 25,000 for the largest training. The resulting set of images is split into two sets each along with their relevant set details, the artificial training set and artificial validation set. The artificial training set consists of a randomly chosen 80% of the images and the validation set, the remaining. The training set is used to train the parameters of the the predictive model. The artificial validation set is used to guide the model hyper-parameter selection. In each case the images undergo an orientation and cropping process before being written to their respective set. In addition, the per pixel mean and variance are computed over the entire training set for use in image normalization during input to the prediction network.

### 3.8.1 Orientation and Cropping

The orientation, and shape of the bag is unknown in an image coming from the plant. In addition, the overall size of the image will be unknown as much of it can be filled with white space. The predictive model can only handle images of a predetermined size. In this case the needed size is 510x380. The problem then is two-fold: how to determine the best place to crop a 510x380 patch from an image and how to evaluate the amount of information retained by any such crop. This is a difficult problem as it has three degrees of freedom each of which has a large parameter space as well as a difficult to compute evaluation.

Considering that the white space in an image is likely to hold very little information, minimizing white pixels in the crop region is used as an approximation of the amount of information in a cropped region. This has the benefit of being relatively

simply approximated and computable in a reasonable amount of time for an individual crop region.

Consider now the problem of choosing where to crop from the point of view of the center of a proposed patch to be taken from the image. This center point can be anywhere along the 2-d plane of the image. In most cases this alone offers over 0.25 million options. Add in that the patch can have any rotation around that center point and there are quickly far too many possibilities to consider in a realistic time frame.

The strategy for finding this optimal crop region begins with getting a reasonable starting point. First the extents of non-white space are found. This consists of finding the lowest, highest, leftmost, and rightmost point that are nonwhite. From these values a centroid is computed. This centroid combined with a 0 radian rotation are used as a starting point and baseline measure of white pixels. From here a search is conducted in two phases, a broad phase and fine phase. The broad phase evaluates each neighbor location spatially at -10,-5, 5, 10 away and rotationally at -15, -10, -5, 5,10,15 degrees. If any of these neighbors provides an improvement over the current proposed patch, the search continues from that new improved best patch. Once no more improvements can be found this way, a new fine search is started from the best found patch in the same fashion, but with spatial and angular increments of -2,-1,1,2. Again when improvements are no longer found the search is terminated and the best found patch is taken. Pseudocode for this process can be found in Algorithm 3.8.1

This greedy neighborhood search is not guaranteed to find the best possible patch, but in practice performs quickly and yields results that look very appropriate via visual inspection.

---

**Algorithm 1 Greedy search for best crop orientation**


---

```

1: procedure ORIENTEDCROP(image, croplims)
2:    $e \leftarrow \text{NonWhiteExtents}(\textit{image})$ 
3:    $p_{\text{best}} \leftarrow \langle \text{Centroid}(e), 0 \rangle$ 
4:   for stride in  $\langle \textit{broad}, \textit{fine} \rangle$  do
5:     while  $\text{Change}(p_{\text{best}})$  do
6:        $N \leftarrow \text{Neighbors}(p_{\text{best}}, \textit{stride})$ 
7:       for  $n$  in  $N$  do
8:         if  $\text{White}(n) < \text{White}(p_{\text{best}})$  then
9:            $p_{\text{best}} \leftarrow n$ 
10:        end if
11:       end for
12:     end while
13:   end for
14:   return  $\text{Crop}(\textit{image}, p_{\text{best}})$ 
15: end procedure

```

---

# CHAPTER 4

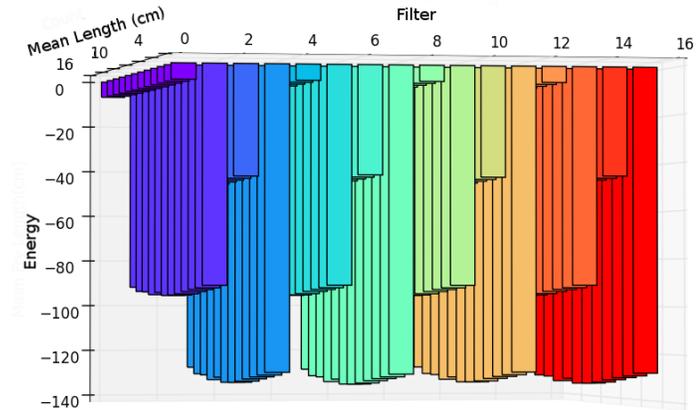
## NETWORK, OBJECTIVE, AND FEATURE SELECTION

This chapter describes the decisions made in relation to the features selected from the input data and the architecture and parameterization of the Deep Convolutional Neural Network prediction model.

### 4.1 Feature Selection

#### 4.1.1 Motivation for Hand Made Features

Although the deep network is capable of feature learning, it is always a good idea to include domain knowledge if it is present. Because a simple tomography image projects the 3-d layout of the product onto a 2-d plane, the intensity in the exposure of any point of an individual product cut in the exposure is dependent on the material above and below it in the bag and is not strictly a function of the cut itself. Therefore a cut can appear in a variety of intensities from very light to near black. The visual definition of an individual cut in an exposure owes much to the fact that there is an intensity gradient at its edge. Taking these facts into account, each image will be augmented with some hand made features that will focus on edge related features combined with contrast normalization pre-processing of the image. This effect is supported by some early exploratory work that shows a correlation between the energy of Gabor filters with small standard deviations convolved over an image and the mean product length in the image. These Gabor filters act much like an oriented edge

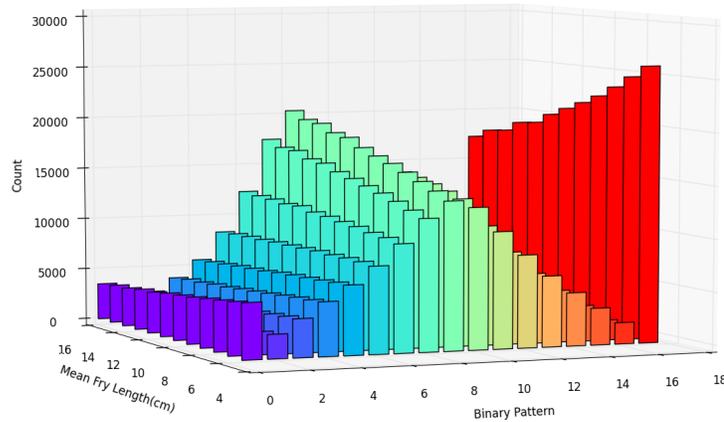


**Figure 4.1:** Cross correlation energy of several Gabor edge filters. These energies are obtained by convolving the filter over whole tomography images and computing the sum of the pointwise energies. This is done for several different image sets each representing a different mean product length.

detection filter. Figure 4.1.1 illustrates an example edge correlation in the exploratory work.

Exploratory work also highlighted a correlation between Local Binary Patterns (LBP) and the mean product length for certain artificial images. For this reason each image is also augmented with a set of LBP features. Figure 4.1.1 shows an example correlation between LBP and mean product length in exploratory work.

There is little risk in adding feature inputs to the network. So long as the features are derived solely from the input image itself there is a guarantee of no information leakage into the system. In addition, with any amount of weight regularization, weights connected to a feature that is of little or no use will drop to negligible levels causing the network to effectively ignore the feature.



**Figure 4.2: Count of different Local Binary Pattern features.** These counts are obtained by computing the LBP at each point in each whole tomography image and computing the sum for each pattern type. This is done for several different image sets each representing a different mean product length.

#### 4.1.2 Edge Features

Each single channel image is augmented with 4 additional channels produced by computing edge features over the image. There are a vast number of edge-like features. Without knowing which are most conducive to this particular problem, it is hard to pick one. Thus a variety are used.

The first two additional channels added to the image are Difference of Gaussian (DOG) features. To compute a DOG feature the image is blurred using a Gaussian blur twice at two different standard deviations. The resulting two images are then subtracted to produce the feature image. This difference approximates a first order derivative over a blurred version of the original image. The blur is useful because it effectively is a feature size selection mechanism. Using a small area Gaussian blur allows fine detail edges to come through. Using a large area blur removes fine

detail features and causes larger scale features to be highlighted. Because the DOG transform is parameterized by the standard deviation used during the Gaussian blur there is freedom in the feature size that is most identified by this transform. Two different parameterizations are used. One uses a difference of a 0.7 std and a 1.2 std Gaussian to pick up fine detail edges. The next uses a difference between a 1.0 std and 2.0 std Gaussian to highlight larger scale edges.

The CenSurE feature detector uses an approximation of the Laplacian of Gaussian filter which uses a second order differential to define edges [1]. This edge detection methodology is adopted here and each image is augmented with a Laplacian of Gaussian (LOG) transform. The LOG feature is similar to the DOG feature in that the original image is blurred using a Gaussian blur, but only once. The Laplacian then computes a second order derivative over the resulting blurred image to produce the feature image. For this application a Gaussian kernel with a standard deviation of 0.7 is used.

The final channel is a Sobel filter convolution. The Sobel edge filter is a simple edge detection filter that simply convolves a 3 by 3 kernel over the image that outputs the sum of a discrete first order derivative of the image in both the vertical and horizontal directions.

### 4.1.3 Pattern Features

Each image is also augmented by an LBP feature set as described by Ojala et al.[23]. The Local Binary Patterns algorithm assigns a single value to each pixel. This value is computed by first finding the sign of the difference between the pixel's intensity and each neighbor pixel's intensity. Each of these signs is then mapped to either a 0 or 1 in the final value at a bit location defined by the neighbor it was computed

from. As a simple example, if each pixel has a neighbor set of 8 pixels, each pixel is assigned an 8 bit value. A pixel which has an intensity greater than all of its neighbors would be assigned 11111111 (255), where a pixel with an intensity greater than half of its neighbors might be assigned 00010111 (23) depending on the pattern of neighbor intensities. In practice the radius and pattern structure of the neighbor pixel set can be controlled with an R and P parameter.

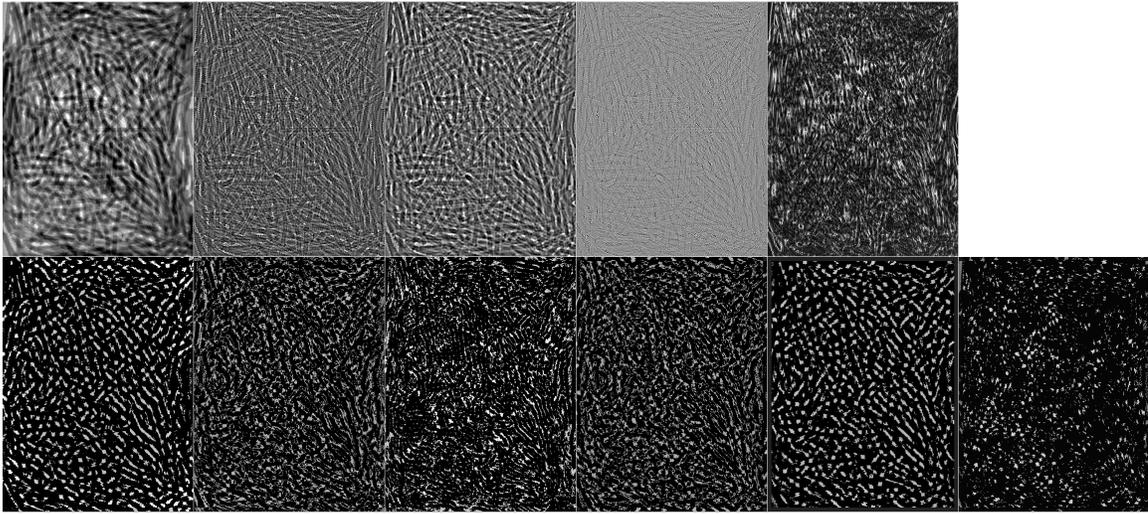
For this application the set of LBP features uses a radius of 8 pixels. Since the expected product width is around 8 pixels this value is expected to extend the feature to the area surrounding an individual cut. It also uses a P value of 4. This value governs the number of rotationally invariant patterns that are checked during feature extraction. This value produces 6 possible output values. Each pixel is labeled with a value in the interval  $[0, 5]$ . In order to be interpreted by the network these are added as 6 additional channels in the image and each is a single pixel label type in a one-hot encoding. This representation is the most direct for consumption by the network but is highly sparse, taking up a lot of memory for the amount of information it contains. The P value of 4 is chosen as a compromise between the resulting size and sparsity of the augmented image and the specificity of the feature set produced.

Figure 4.1.3 shows an example of the channels in an image augmented with the above described features.

## 4.2 Objective Selection

### 4.2.1 Network Outputs

One of the major decisions in defining the objective function, is to decide what the outputs of the network will be. In some cases this is fixed by the problem at hand.



**Figure 4.3:** An image example input into the network. The images in order are: top row: original, fine DOG, broad DOG, Laplacian, Sobel. bottom row: LBP features 0-5. Note that the augmented image has been normalized by subtracting the training set mean and dividing by the training set variance. Each channel has been normalized to cover the full spectrum for the purposes of print.

Here there is a great deal of freedom in defining these outputs. Very specifically, the goal is to determine whether a bag is acceptable or unacceptable as defined by some specification. The naive approach would be to treat this as a binary classification problem. Indeed, early in experimentation with the problem this was the approach. It was however, not fruitful. The most plausible hypothesis for this is that the images obscure some of the information regarding the complete set of cuts. Therefore any bag near the threshold for acceptance may result in an image that indicates that it is on the other side of the threshold. To the network, this comes across as a mislabeled image. When the labels are binary that is a drastic difference. The way around this is to treat this as a regression problem. This way a bag containing say 23% short cuts that produces an image that indicating 25% short cuts will not produce as large a misleading error signal, and indeed still drive the network to roughly the correct

determination. This has the added benefit of providing a richer amount of output information. With regression values, it is possible to perform statistical analysis on the outputs and get an idea of the degree to which a bag is acceptable or not.

The set of outputs that is in the end used is the mean product length, % product from 0" to 2" length, % product from 2" to 3" length, % product from 3" to 4" length, and % product with length 4" and above. The mean product length is chosen since it was found to correlate well with many image features early in the exploratory work and continued to be a measure with fairly low error. The percentage measures are chosen because they are interpretable measures independent of the number of cuts in the bag and are consistent with measures used commonly in the factory production line. They are also chosen to be mutually exclusive to enable an objective term to drive them to sum to 100%. Without this, it was found in early training of networks that the sum of a set of mutually exclusive percentages would most often have a sum above 90% but could be as low as 40%. The network itself has no notion of percentage, so this is another simple opportunity to add in some prior knowledge about the problem.

#### **4.2.2 Objective Function**

The network is trained by backpropagating error gradients. This necessitates the choice of an objective function to derive these errors and gradients from. The objective function used here is a weighted sum of independent objective functions designed to drive certain desired network criteria.

The most obvious part of any objective function is the error associated with the difference between the predicted and actual label of an example. It is helpful to recall here that the examples come from two groups, the historical and non-historical. The

historical examples are tightly clustered, whereas the non-historical have a broad spread. To get a value for regression error two terms were used. First the Mean Squared Error (MSE) is computed. This term is insignificant for small errors but rapidly rises for larger ones. It is used to keep the network from drastically mislabeling any examples. This is most useful for the non-historical outliers which are very small in number. Because they are so small in number they naturally provide a very small portion of the error signal as the network will attempt to model the much larger set of historical examples. The MSE keeps the error on these few outliers at reasonable levels. The second error term used is the Mean Absolute Error (MAE). This term scales linearly with error and is equally sensitive to small errors as to large ones. This term is included to drive the network to minimize the error of the large numbers of historical examples. With the low sensitivity to small errors of only a MSE the network can achieve low objective error while still having substantial error on a large number of the arguable more important historical examples.

In a vein similar to the MSE, MAE combination used above for error, a combination of L1 and L2 weight decay terms are used. L1 computes a degree 1 norm over the parameters in the network. This amounts to calculating a MAE where each parameter is treated as an error. Similarly, L2 computes a degree 2 norm over the parameters, amounting to essentially a MSE with parameters interpreted as errors. These both perform the function of regularization on the network so that it does not model the intricate details of the training set so well that it does not generalize to the validation and test sets. Much like the behavior of the MSE and MAE used for labeling errors, the L2 prevents the network from driving any parameters extremely high, thereby emphasising the importance of any one feature too much. The L1 term provides a weight reducing pressure regardless of the magnitude of the weight. This

has the effect of pushing weights to 0 unless the weight has some sort of positive gradient value. This is convenient for elimination of features that are not useful for prediction, as they will not have sufficient positive gradient values and eventually become ignored by the network.

The final term added to the objective function is a MAE derived from the difference of the sum of the bin percentages from 1.0. This term simply drives the network to ensure that those percentages consistently sum to 1.

### 4.3 Model Architecture Selection

The predictive model, a deep convolution network, was developed with several constraints in mind. It must be fast enough to give predictions in the time required by the production line hardware while also being capable of running on reasonably available hardware. To achieve this the network takes an architecture similar to that of VGGnet by Simonyan et al. [26]. This network consists of a series of interleaved 3x3 convolution filters and max-pooling layers. These two component choices provide two substantial benefits.

First of these is speed. A convolution operation without striding runs in  $O(d^2 f^2)$  time where  $d$  is the size of the image on a side in pixels and  $f$  is the size of the convolution filter on a side in pixels. Since the image dimensionality is more or less fixed, the only opportunities to reduce the computation time are to stride or reduce the filter size. Striding means that the network is not applying the filter at every location in the image. This is acceptable in images with high resolution since an edge or other feature's width can span several pixels. With images at only 29  $\frac{px}{inch}$  this is not likely the case, and striding is likely to cause the filters to miss some of those

features. This leaves reduced filter size as the only viable option.

The second benefit of this architecture is the size of the embedding. An image entered into the network must be represented as a set of activations in each of the filters. Using a large number of filters requires storing a large number of these activations. Because the dimensionality of a 3x3 filter is small there are fewer possible filter states and thus fewer filters required to reasonably span those states. Note that this benefit are not for free. The tradeoff here is that a deeper network is needed. Recent work by Lin and Tegmark [17] suggests that this tradeoff is a good one to take as deep networks create a hierarchical representation that is more suited to the types of data that exist in the world courtesy of the hierarchies present in the laws of physics. The second half of reducing the embedding size is the use of max-pooling. This technique takes a 2x2 patch of activations and collapses them to a single value taken as the max in the 2x2 pool. This reduces the embedding spatial dimensionality by 4x. This also has the side benefit of making the filters more coherent [24].

The details of the network architecture can be see in table A and table A in section A. The depths for each of the layers is chosen to restrict the model capacity but still provide the needed modeling power to achieve performance. These have mostly been selected through trial and error. In that process it could be seen that large depths resulted in very low training error with rising validation error, and small depths resulted in substantial error in both sets that did not decrease with more training time. These depths represent the middle ground between those two extremes where training error and validation error both drop to a stable value over the course of training time. Every one of the non pooling layers uses a leaky rectified linear unit (ReLU) layer with a slope of 0.001 in the leak region. The leaky ReLU is a function defined as follows

$$f(x) = \begin{cases} x & x \geq 0 \\ kx & x < 0 \end{cases}$$

where  $k$  is some small constant  $k \ll 1$ . This activation function has several benefits over traditional activation functions. In the positive region it is linear and not prone to saturation problems like sigmoid or arctan activation functions. During back propagation the full gradient value passes through the function unaltered, thereby avoiding the problems in deep networks of disappearing gradient or gradient explosion [14]. It also has a range that is unlimited for the purposes of the network. This allows it to work well in a regression setting where real valued outputs are needed. In the negative domain, it provides the non-linearity with respect to the positive domain, that is needed for a network to function properly. Because of the  $k$  constant, there is still the opportunity for gradient information to pass through thus avoiding the dead pathways that a standard ReLU can cause.

#### 4.4 MetaModel

Each tomography image offers only a limited view of the content of the bag, so there is likely a substantial uncertainty surrounding any classification made from such an image. If one assumes that the underlying distribution of lengths coming from the line changes slowly, then it can also be assumed that for a small set of consecutive bags, the distribution will be near stationary. Each image offers some incomplete information regarding this distribution and can be interpreted as a sample. Using such a statistical interpretation, increasing the sample size will decrease the uncertainty in the estimation. In this case a simple median is found over the set of deep convolution net predictions for batches of 33 images to reduce that uncertainty. 33 images is used

partially because it aligns with other measures taken in the plant in 33 bag intervals. This number is convenient however since it is just above the commonly accepted sample size of 30 used as a rule of thumb for accepting estimates based on Student's T statistic for mean values.

Both the mean and median of 33 bags were looked at in the validation sets, where the median gave slightly better results. Compared to using individual bag instance predictions, the expected error and standard deviation of the prediction errors was as low as half for some prediction types using the median of 33 instances. By using this method, there is an assumption of stationarity in the distribution of product lengths over brief periods of time as the bags come down the factory line. This assumption has not been investigated yet, partially due to the difficulty in collecting real labeled samples.

# CHAPTER 5

## PERFORMANCE EVALUATION

### 5.1 Data Sets

Four data sets are used to evaluate the performance of the deep network in predicting product length statistics. These sets are derived from the combinations of historical vs mixed crossed with the presence of what will be termed "realism factors". Mixed refers to the combined sets of historical and non-historical data. The details of historical vs non-historical are described in 3.3.1 and 3.3.2. The realism axis of variation is described in the next two sections.

#### 5.1.1 Without Realism Factors

The images in these sets are created in a sort of ideal scenario where the bag is always level and well formed, and the inspection machine functions as expected. Granted, this is not a realistic scenario but gives some level of performance that could be achieved if enough work was done to create those circumstances. For these sets of images, the simulation used no pitch or roll when settling the meshes into the container. The overexposure effect described in 3.6.2 is not simulated. Table 5.1.2 details the differences between this set of images and the Realistic ones.

#### 5.1.2 With Realism Factors

The images in the Realistic set are intended to model the sort of variations seen in the plant production line. In this scenario some bags are tilted along various axes and are occasionally misshapen. In addition there is a chance of overexposure effect,

**Table 5.1: Details of the differences in parameters used between the ideal and realistic training sets.**

Value	No Realism	Realism
pitch $(\mu, \sigma)$	(0,0)	(0,0.18)
roll $(\mu, \sigma)$	(0,0)	(0,0.18)
yaw $(\mu, \sigma)$	(0,0)	(0,0.2)
overexposure $(\mu, \sigma)$	(0,0)	(0,0.1)

likely an artifact of the tomography equipment operation. Creating these effects in the artificial images is a much harder task. In many cases the root cause of the effects seen in the real images is unknown and can only be guessed at or approximated. This being the case it is difficult to know how to parameterize such an approximation. The "fit" of the artificial data to the real images can only be known after they are created and relevant statistics can be gathered. This is an expensive process in terms of time. For this reason the values used here represent the current state of iteratively refined approximations, not necessarily the best that can be achieved nor an authoritative set. Again table 5.1.2 shows the values used to produce the images with realism factors.

## 5.2 Baselines

To the best of our knowledge, there is no prior work to be found in the area of automated length measurements in tomography of aggregate materials. Because of this there is no baseline performance measure to compare a deep convolutional network against. Though not necessarily needed in terms of evaluating a CNN for a particular business process, it is convenient to see where such a method falls in comparison to others. Several proposed baselines are proposed here with the context of use in the production line in mind.

### 5.2.1 Immediately Feasible

Two baseline methods are immediately feasible in the plant currently. The first of these is simply assuming that each bag matches the historical mean statistics. This method at first seems laughable, and as a true business process it is. This baseline prediction is possible without any knowledge of the actual example that a prediction is required for. It simply represents the ability to learn an expectation for a population, a bias. However, because of this fact it represents a baseline measure that any machine learning system should be able to match or outperform. It is this baseline that essentially represents the null hypothesis for this thesis. If a tomography image does not provide enough information to render an estimate of lengths, it is at best this performance that should emerge regardless of modeling technique employed.

Another immediately feasible method is that currently employed by the production line, that of sampling. Sampling is a common statistical practice when information about an entire population is desired but the entire population cannot be inspected. In this case the sampling is a single bag sample taken from the population. For any of the generated, or labeled real data sets there is complete information available over the sample population making it easy to estimate the expected error for this type of method. To do this a simple average is taken over the errors between each bag's statistics to every other bag's statistics. This average represents the expected error using a single sampled bag's statistics as a proxy for any other bag in the sample population.

### 5.2.2 Convolutional Neural Network Alternatives

Two alternatives to a deep convolution net are considered as baselines as well, a linear regression, and a Support Vector Regression (SVR) machine. These two methods attempt to represent machine learning methods that are generally accepted and require less training effort than a CNN. This also provides two baseline measures that utilize the images from the production line. In both cases, these methods work on a flat feature set. That is to say that they do not recognize spatial relationships in the features that are given as input. Therefore providing the raw pixel values as input is not expected to give good results in either case. To address this, a set of flat features is taken from each of the images to be used in the linear regression and SVM. For the reasons described in 4.1 these features focus on edges and local binary patterns.

In particular a set of HOG features is derived. HOG is an acronym for Histogram of Oriented Gradients, a method developed by Dalal et al. [7]. This method logically partitions the image into tiles. For each tile a histogram of edge or gradient directions is accumulated. The histogram results for each tile are accumulated to form a flat histogram for the entire image. For this application tiles of dimension 24x24 are used and 7 gradient orientations collected.

A histogram of LBP features is concatenated with the HOG features to form the feature set used for each image. In early exploratory work some features exhibited a quadratic relationship with mean product length. For this reason, the feature set used here is augmented with the set of the square of each feature

The first of these, the linear regression is a simple sum of the products of the image features and learned parameters. Its formulation for a given example is as follows

$$y_k = \sum_i p_{ki} X_i$$

where  $y_k$  is the  $k^{\text{th}}$  prediction in the prediction vector and  $X_i$  is the  $i^{\text{th}}$  feature for the example.

SVR is an extension to the well regarded Support Vector Machine proposed by Smola and Vapnik [27]. Assuming  $n$  examples  $x_i$  and  $n$  labels  $y_i$ , the standard SVM formulation is

$$\text{minimize } \frac{1}{2} |w|^2$$

over all  $w \in R^d$  and  $b \in R$  subject to,

$$y_i(\langle w, x_i \rangle + b) - 1 \geq 0 \text{ for all } i.$$

which attempts to separate a set of data with binary class labels. It does so by attempting to learn a weight matrix that when applied to the data will produce a margin around the separation hyperplane. The SVR substitutes the SVM constraint with

$$\begin{aligned} \langle w, x_i \rangle + b - y_i &\leq \epsilon \text{ for all } i. \\ -\langle w, x_i \rangle - b + y_i &\leq \epsilon \text{ for all } i. \end{aligned}$$

This constraint attempts to learn a weight matrix that when applied to the data will constrain the prediction error by a fixed amount. The optimization of an SVR depends on a kernel function that outputs a real value for any pair of examples.

In part, the expressive power of the SVR depends on the choice of kernel function. For this application the radial basis function kernel is used which is known to result in a quite expressive model.

### 5.3 Evaluation

Each of the baseline prediction methods and each of the machine learning models have been trained on artificial images spanning the combinations of historical and realism factors. The details of each of these data sets, and models are discussed in section 5.1 and section 5.2. The details regarding the performances are listed in B. One important detail here is that although a mixed, historical and non-historical, data set is used with both realism and no realism, those two mixed sets are different in nature. The mixed set without realism derives mostly from the scaled flat distribution whereas the mixed set with realism derives more from the iterative flat distribution, making it difficult to compare results from those two columns.

A general trend to note in the performance details is that the SVR model suffered from substantial overfitting on the training set. Its prediction errors on that set were minimal, yet for unseen data, the output remained mostly invariant. For artificial data that output appeared to be the data mean. For real images the output appeared to be skewed but still mostly invariant. For this reason, the expected errors for the SVR in the artificial image performance are nearly the same as the "Guess the Mean" baseline. The choice of the radial basis kernel function given the number of training examples is most likely the cause of this problem. Either a less expressive kernel would need to be used or more training data gathered. Regardless, the SVR results will not be given much further discussion.

Another general trend that arises is that a single bag sampling strategy does not allow for a very good expected error for any of the data sets. This is due simply to the variance of the product lengths. A simple statistical solution to this problem that may not be as simple in a true production environment is simply to take more samples. Further discussion on this can be found in section 6.3.7.

### **5.3.1 Realism**

Adding the types of realism factors shown in table 5.1.2 seems to have a slight detrimental effect on the capacity of a model to predict lengths on unseen examples. This can be seen by comparing the performance of the linear regression model for the “ $\% < 2$ ” and mean length outputs between the two historical data sets. In these cases the label types and historical nature of the data remained the same. Where realism had made prediction more difficult for the artificial data, it greatly improved the capacity for the linear regression to generalize to the real data.

### **5.3.2 Historical Accuracy**

More of an effect on prediction capacity is seen by varying the historical accuracy of the data. For the mixed data sets, that represent a wide variety of product lengths, the machine learning models are capable of making length estimates that are substantially better than a mean guess or a single bag sampling, with the CNN models achieving the top performance. Reducing the variability of the product lengths down to what is historically accurate changes the story. In this context the improvements in prediction that can be made over a mean value guess are slight. The precision required to make a better prediction than a mean guess is possibly near or more than the amount of information present in the image. For artificial images the linear regression model

does slightly better than a mean guess, but that performance does not hold in the context of the real images.

### 5.3.3 Findings

In the best case, a plant has high quality images with a product length distribution that is highly varying, much like those in the mixed without realism data sets. In this scenario there is much to be gained by using a learned model to predict those lengths. Historical measures, nor small sample sizes can account for the high variance and the image is ripe with information.

At the other end of the spectrum are either noisy, distorted images with poor product orientation, or product that has a very narrow distribution. In the case of noisy or distorted images it will be difficult to extract the desired prediction because the needed information is not strong enough to rise above the background effects or has been lost altogether. In the case of product with a very narrow distribution, simple knowledge of the historical mean provides an estimate with a low expected error, one that is difficult to best with limited information in an image. The important question is at what point does it become advantageous to invest in a machine learning approach. This work has shown that the axis of expected length distribution should be a primary factor in that decision. Models predicting on the mixed data have better performance than a mean guess or single sampling regardless of the realism factors introduced. On the other end of the spectrum, complete removal of realism factors, a near perfect image, in the historical set led to only very slight advantages over guessing the mean. In fact, the small magnitude of that advantage suggests that the limit of precision in predicting these lengths from information in an image via a learned model might be near where the historical output lies.

# CHAPTER 6

## CONCLUSIONS

### 6.1 Current State and Future Directions

We have shown with this work that a grey scale image produced from an radiation tomography process on fried and bagged batonnet cut potato does indeed contain enough information to produce estimates of the lengths of the cuts within. More important is that the noise rate in those estimates is dependent on several factors. The noise in the estimates tends to rise as variability is introduced into the process of producing the images. This can take many forms. This work shows that at least the orientation of the bag in the roll and pitch directions can decrease estimate quality. The sensitivity and stability of the radiation detector can also degrade the types of length inferences that are possible from the resulting images. Additionally the quality of the prediction tends to decrease with the diversity of the length distributions in the data the model is trained on.

This work is an investigative work into the feasibility of predicting length statistics of product using radiation tomography appliances. Having shown that there is an opportunity to take advantage of the information contained in such an image, what steps are needed to realize the potential? This question can be viewed from two opposing perspectives. On one side of the token, work could be done to improve the fit of the artificial data and modeling to the real processes in the plant. On the other side, the processes in the plant could be made to produce data more conducive to modeling. The optimal path likely lies in a compromise on both sides. The details of these two directions are discussed in the next two sections.

## 6.2 Improvements To Modeling

Several areas of further investigation would likely improve the modeling pipeline for this particular problem.

The choice of outputs for the deep neural network remains an open question. This choice walks a line between simplicity of prediction and usefulness in making business decisions. Choosing percentage of cuts per length bin as an output as discussed in section 4.2.1 is intentionally biased toward usefulness. It is possible that an output such as a raw count rather than a percentage would allow for more accurate predictions as it is a simpler function not relying on the total number of cuts in the bag. The number of possible output combinations is limitless and warrants some further investigation.

The disparity between the predictions on artificial images versus real images suggests that there is a mismatch between the two. Work needs to be done to determine what those differences are and alter the artificial data pipeline to produce images that more closely reflect the real ones. In terms of training on artificial data and generating predictions on real data this is likely the area of improvement that would prove very beneficial. This line of investigation would likely necessitate the creation of additional group statistics to supplement those described in section 3.7 to guide the revision of the pipeline.

One likely influence on the difficulty of matching the artificial images to the real ones is the uncertainty surrounding the characteristics of the tomography machine. As discussed in sections 3.6.1 and 3.6.2 some of the aspects of the machine generating the images are unknown and must be inferred. Worse yet, some of them must be set as adjustable parameters to the image modeling process and driven by feedback

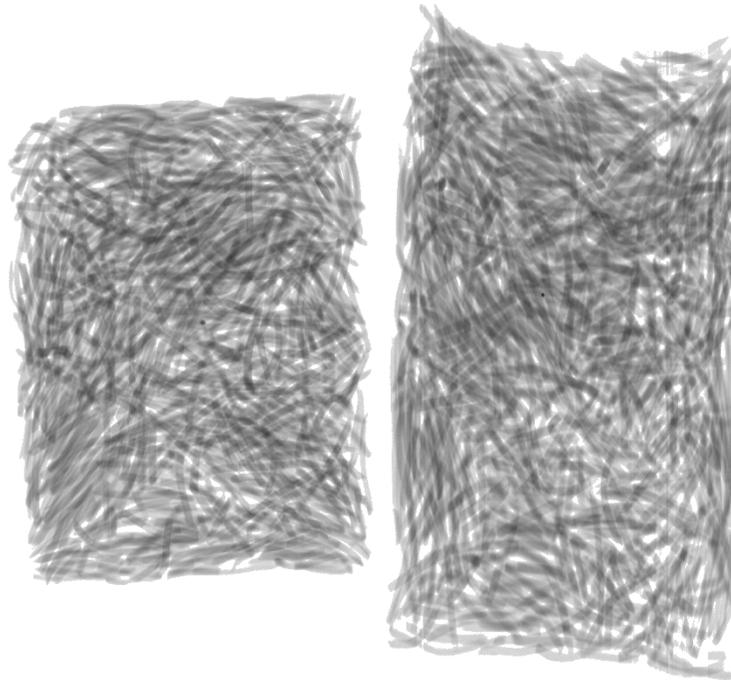
from statistics describing the similarity between artificial and real images. Ideally the ultimate outcome would be an artificial pipeline that correctly models the real images. However, establishing this fact depends on choosing the correct statistics over the outcome. Doing this is difficult, and it is never clear that it has been done well enough. Directly getting the qualities of the tomography machine and using them in the artificial image pipeline would potentially be easier and certainly less error prone.

### **6.3 Improvements To Preparation**

One of the more difficult parts of producing artificial images for use in modeling is the fact that there are complex physical effects being realized in the real images. While efforts in improving the "fit" of the artificial images is likely to help, there is also opportunity in reducing the amount of variation in the way images are produced. The size of the bags used to hold the product is roughly fixed. However in the images from the plant there are several variations in the dimensions of the bag image. The sources of these variations are likely many fold, but some prominent effects rise to the surface.

#### **6.3.1 Image Stretch**

The most prominent of these in relation to the task at hand is that of a stretching effect. This effect is isolated to the direction of conveyor belt travel and is likely due to a slowing of the belt. As the belt slows the resolution in that direction increases since the detector has the opportunity to take more samples per inch in that direction. Figure 6.3.1 shows an example of this effect. This variation is particularly concerning in this case because it affects the lengths of any cuts in that orientation. Since it is



**Figure 6.1:** An image example of image stretch likely due to conveyor belt slowing. The image on the left is a normal real image. On the right is a real image displaying stretch.

exactly length that is to be predicted, this is likely to cause error in the estimate. It is possible that normalizing the size of the bag outline in the images via scaling could help, but the most effective initial step to address this would be to attempt to control the belt speed.

### **6.3.2 Image Resolution**

This work has shown that in some cases a tomography image does contain enough information to recover statistical properties about the product in question. Since information content is a critical property for this work it is important to ensure that the images used have maximal information. To this end, there are three areas where improvements in resolution could be realized.

The spatial resolution of the images is a source of concern. In this case the images have a resolution of 29 pixels per inch providing product widths of about 8 pixels. This low resolution causes the location of edges to be imprecise. This low spatial resolution also means that flux is being collected in blocks of  $\frac{1}{29}$ ". For illustrative purposes, consider a sharp edge between white and black that lies in the center of one of these blocks. Although the edge should be a sharp contrast, the area in the block is half black and half white causing the pixel to be rendered as a middle grey, hardly a sharp edge. This sort of effect causes edge intensity to be lessened throughout the image.

The intensity resolution of the images is also a problem area. The images used in this work are encoded as 8 bit values yielding 256 different gray scale values. However, the values 225 to 254 never occur in real images along with values below about 60 as is shown in figure 3.7.1. Together this constitutes about 35% of the possible intensity values that are never used, effectively reducing the intensity resolution by the same amount. Edges in an image are detectable as intensity gradients. With reduced intensity resolution, edge detection capacity is likewise reduced.

Finally, improvements could be made in the precision with which the radiation intensity is resolved. Section 3.6.2 and Figure 3.6.2 detail the problems with noise seen in the sample real images. This noise serves generally to increase entropy and reduce information content in the images, but also has compounding effects with the resolutions described above. In some cases the noise in the detection can make an edge appear displaced by a pixel. When the anticipated width is only 8 pixels to begin with, that single pixel displacement constitutes a large difference. Similarly, when the effective intensity range is only 65% of the total range, noise fluctuations in intensity have a magnified effect. Though not the sole reason, this likely accounts

for some of the reduced predictive power observed in the results when training with realistic training data.

### 6.3.3 Overexposure

The overexposure effects described in 3.6.2 are also an area where improvements could be made in the imaging process. Exploring the cause and potential solution to this effect offers two benefits. First, the whitening effect appears to sometimes be strong enough to completely eliminate certain product cuts in the image, particularly those that are isolated or in areas where the layering is thin. Obviously, eliminating the signal from any individual cut will lower the amount of information content in the image and therefore the ability to render an accurate prediction. It is particularly harmful in this case because the isolated or thin area cuts are likely the most informative in a regular exposure since the signal at the detector from those cuts is not in contention with the signals from overlapping cuts.

Secondly, CNN's along with many other machine learning techniques, tend to give better results when the inputs are normalized. Even if it does not wholly eliminate signals in the image, the variability in brightness caused by the intermittent overexposure effect in turn causes variability in the prediction model. To a certain degree these models can be robust to such noise, or simply be trained to integrate that into the model. However, taking a factor like this into account is not easy. Modeling it in the artificial images is a tedious task, complicated by the need to measure how well it has been modeled which itself is not easy. The overexposure effect displayed in the real images also appears to mostly occur on the right side of the images, making the images asymmetrical. Given an expectation of symmetrical images, one training image can be flipped vertically, horizontally, and in both directions to yield 3

additional training images. This 4 times increase in training data is nearly free and is an immediate way to combat the potential overfit of a model to a small data set. The asymmetry of the overexposure in this case causes this technique not to be feasible.

#### **6.3.4 Bag Orientation**

For similar reasons, it would be beneficial to make efforts to normalize the shape and layout of the bag on the conveyor belt. Though the interior region of the bags is generally filled with product, often the edges are unpredictably shaped. Sometimes a corner appears folded or a whole bag twisted or bent in a curve. Much of the time this results in an area of the resulting image being white. Ultimately this white region causes the prediction model to suffer from the same effects as the overexposure problem. Certainly it can't be expected that every bag is a perfectly formed rectangle, but likely a little effort in aligning the bags and possibly vibrating them to fill the corners would go a long way toward normalizing their appearance upon presentation to the prediction model. In the event that the natural alignments of the bags are currently biased in an asymmetrical way, normalizing the alignment would make possible the image flipping data augmentation strategy discussed prior.

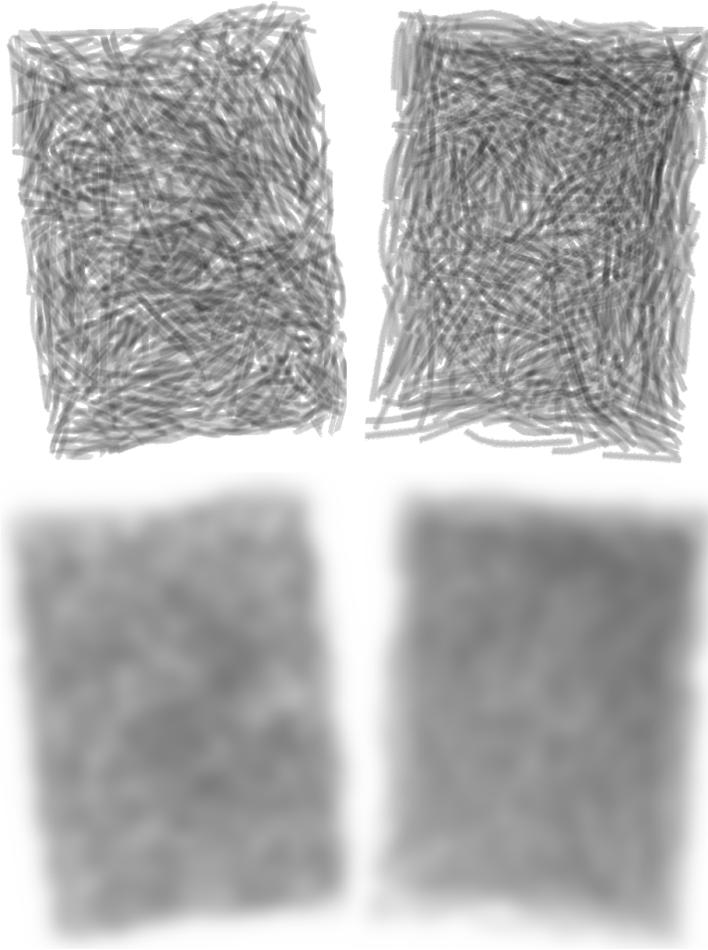
#### **6.3.5 Product Dispersion**

A side benefit of vibrating the bags includes potentially evening out the interior layout of the product. Many of the real bags display a noise on the scale of inches. This is to say that in many of the real tomographic images there will be a region of perhaps 2"x2" that has white area and is generally lightly colored near other similarly sized regions that are dark. This is very clearly an effect of the arrangement of the product in the bag, and one that the artificial simulator does not currently capture. Figure

6.3.5 shows this effect pretty clearly. Though the whites and greys on the scale of an individual cut's width are similar between both real and artificial, the blurred version of each image reveals that despite showing a gradient darkening toward the upper right corner, the artificial one is more homogeneous on a larger scale where the real one is not. Again, removing variations like this is only to the benefit of any predictive model compounded by the fact that it is not clear how this effect would be created in the simulator.

### **6.3.6 Data Collection**

Machine learning models are capable of making use of near limitless data, and for models with large numbers of parameters that large amount of data is not just helpful but needed. Artificial data generation can likely fill much of that gap, however it is not likely that a model conditioned on artificial data alone will achieve maximal performance. It is still important to expose a model in training to data taken from the process of interest. For this, many images are needed. For context, the top models in the well known yearly Image-net image recognition competition utilize 1.2 million labeled images of size 256x256. The images used here are sized 510x382, 3 times as many pixels per image. Obviously, it is not possible to collect so many real labeled images from a production line. However, approaching 10s of thousands might be enough to condition a model for desired results. The difficulty in doing this lies in ensuring that those images are indeed taken from the process of interest. Assume that several thousand images are collected, and the tomography machine is re-calibrated. Variations in the training set are often beneficial, and there are ways to adjust outdated data if the calibration changes made are well known. However, those thousand images might be in jeopardy of being unusable since the process no



**Figure 6.2:** An example of large scale variation in real images. The images on the left are the same real image. On the right are an artificial one. The blurred version of these images on the bottom illustrates the brightness variations in the real image compared to the lesser variation in the artificial one.

longer matches the one they were taken from. Therefore this task must be taken on carefully in small amounts to guide changes. Only after changes have been finalized can mass collection take place.

### **6.3.7 The Value in Sampling**

Regardless of the radiation based processing done, it is advisable that a bag sampling strategy coupled with some analytical statistics be a part of any final solution. Superficially no model will predict better lengths than can be obtained by a sampling process that inspects a bag's contents. However, combined with the information from sampling in a statistical framework, more useful inferences might be possible. One of the more encouraging observations in the historical data toward this end is the fact that across the bin counts the product distributions take on a shape very reminiscent of the Normal distribution. The vast amounts of theory surrounding the Normal distribution provides a lot of inferential power in sampling from a process known to be of that distribution. Establishing the degree to which this is true for the product distributions and structuring a sampling procedure around that might hold great potential.

Another related area to investigate is that of aggregation. A substantial amount of the prediction noise associated with variations in the image producing process can be overcome simply by taking the median of 33 examples as described in section 4.4. However, it is not known to what degree 33 consecutive bags on the production line can be expected to have the same distribution. At minimum that alone needs to be investigated. Once more is known about the nature of consecutive bags on the production line, it is possible a more sophisticated statistical model could be used to eliminate noise in the predictions coming from the primary predictive model.

## REFERENCES

- [1] Motilal Agrawal, Kurt Konolige, and Morten Rufus Blas. Censure: Center surround extremas for realtime feature detection and matching. In *European Conference on Computer Vision*, pages 102–115. Springer, 2008.
- [2] Blender Online Community. *Blender - a 3D modelling and rendering package*. Blender Foundation, Blender Institute, Amsterdam, 2016.
- [3] Saxby Brown, Dale L Bailey, Kathy Willowson, and Clive Baldock. Investigation of the relationship between linear attenuation coefficients and ct hounsfield units using radionuclides for spect. *Applied Radiation and Isotopes*, 66(9):1206–1212, 2008.
- [4] Bullet. *Bullet Physics Library*, 2016.
- [5] Pictiaw Chen and Z Sun. A review of non-destructive methods for quality evaluation and sorting of agricultural products. *Journal of Agricultural Engineering Research*, 49:85–98, 1991. Need to purchase or get through BSU.
- [6] Zvi Cohen, Steven E Seltzer, Michael A Davis, and Robert N Hanson. Iodinated starch particles: new contrast material for computed tomography of the liver. *Journal of computer assisted tomography*, 5(6):843–846, 1981.
- [7] Navneet Dalal and Bill Triggs. Histograms of oriented gradients for human detection. In *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)*, volume 1, pages 886–893. IEEE, 2005.
- [8] Dr Franck P. Vidal. *gVirtualXRay*, 2016.
- [9] EE Finney and KH Norris. X-ray images of hollow heart potatoes in water. *American Potato Journal*, 50(1):1–8, 1973.
- [10] Ronald P Haff and Natsuko Toyofuku. X-ray detection of defects and contaminants in the food industry. *Sensing and Instrumentation for Food Quality and Safety*, 2(4):262–273, 2008. Describes the goals, illustrates some examples, and highlights the difficulties of automated flaw detection in foods using x-rays. Describes basic X-Ray technology and limitations. Suggests that in order to get real-time detection at high speed a low energy High current x-ray system should be used with material close to detector and far from emitter.
- [11] Jan G Hildell, UR Nyman, ST Norlindh, SF Hellsten, and PB Stenberg. New intravesical contrast medium for ct: preliminary studies with arachis (peanut) oil. *American Journal of Roentgenology*, 137(4):777–780, 1981.

- [12] Kurt Hornik, Maxwell Stinchcombe, and Halbert White. Multilayer feedforward networks are universal approximators. *Neural networks*, 2(5):359–366, 1989.
- [13] Koray Kavukcuoglu, Pierre Sermanet, Y-Lan Boureau, Karol Gregor, Michaël Mathieu, and Yann L Cun. Learning convolutional feature hierarchies for visual recognition. In *Advances in neural information processing systems*, pages 1090–1098, 2010.
- [14] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*, pages 1097–1105, 2012.
- [15] B Boser Le Cun, John S Denker, D Henderson, Richard E Howard, W Hubbard, and Lawrence D Jackel. Handwritten digit recognition with a back-propagation network. In *Advances in neural information processing systems*. Citeseer, 1990.
- [16] Yann LeCun, Koray Kavukcuoglu, Clément Farabet, et al. Convolutional networks and applications in vision. In *ISCAS*, pages 253–256, 2010.
- [17] Henry W Lin and Max Tegmark. Why does deep and cheap learning work so well? *arXiv preprint arXiv:1608.08225*, 2016.
- [18] Ehsan Lotfi, M Yaghoobi, and HR Pourreza. A new approach for automatic quality control of fried potatoes using machine learning. *Islamic Azad University, Mashad Branch, Ferdowsi University of Mashad, Khorasan Research Center for Technology Development*, 2009.
- [19] M Mellema. Mechanism and reduction of fat uptake in deep-fat fried foods. *Trends in food science & technology*, 14(9):364–373, 2003.
- [20] Domingo Mery and Miguel Angel Berti. Automatic detection of welding defects using texture features. *Insight-Non-Destructive Testing and Condition Monitoring*, 45(10):676–681, 2003.
- [21] Taghi Miri, S Bakalis, SD Bhima, and P Fryer. Use of x-ray micro-ct to characterize structure phenomena during frying. In *13th World Congress of Food Science & Technology 2006*, pages 23–23, 2006.
- [22] Nuri N Mohsenin. *Electromagnetic radiation properties of foods and agricultural products*. CRC Press, 1984.
- [23] Timo Ojala, Matti Pietikainen, and Topi Maenpaa. Multiresolution gray-scale and rotation invariant texture classification with local binary patterns. *IEEE Transactions on pattern analysis and machine intelligence*, 24(7):971–987, 2002.

- [24] Dominik Scherer, Andreas Müller, and Sven Behnke. Evaluation of pooling operations in convolutional architectures for object recognition. In *International Conference on Artificial Neural Networks*, pages 92–101. Springer, 2010.
- [25] Muhammad A Shahin, EW Tollner, RD Gitaitis, DR Sumner, and BW Maw. Classification of sweet onions based on internal defects using image processing and neural network techniques. *Transactions of the ASAE*, 45(5):1613, 2002.
- [26] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014.
- [27] Alex Smola and Vladimir Vapnik. Support vector regression machines. *Advances in neural information processing systems*, 9:155–161, 1997.
- [28] Stephen M Seltzer. *X-Ray Mass Attenuation Coefficients*. NIST Physical Measurement Laboratory, 2016.
- [29] Akira Tsuchiyama, Kentaro Uesugi, Tsukasa Nakano, and Susumu Ikeda. Quantitative evaluation of attenuation contrast of x-ray computed tomography images using monochromatized beams. *American Mineralogist*, 90(1):132–142, 2005.

# MODEL ARCHITECTURE DETAILS

**Table A.1: Details of the network architecture used for training on both data sets with realism and the mixed data set without realism.**

Layer Type	Spatial Dims	Depth	Embedding Dims
Convolution 1	3x3	24	508x380x24
Pooling	2x2	–	254x190x24
Convolution 2	3x3	32	252x188x32
Pooling	2x2	–	126x94x32
Convolution 3	3x3	48	124x92x48
Pooling	2x2	–	62x46x48
Convolution 4	3x3	64	60x44x64
Pooling	2x2	–	30x22x64
Convolution 5	3x3	80	28x20x80
Pooling	2x2	–	14x10x80
Convolution 6	3x3	96	12x8x96
Pooling	2x2	–	6x4x96
Convolution 7	3x3	96	4x2x96
Pooling	2x2	–	2x1x96
Fully Connected 1	–	128	128
Fully Connected 2	–	128	128
Fully Connected 3	–	5	5

**Table A.2: Details of the network architecture used for training on the historical data set without realism.**

Layer Type	Spatial Dims	Depth	Embedding Dims
Convolution 1	3x3	32	508x380x24
Pooling	2x2	–	254x190x24
Convolution 2	3x3	64	252x188x32
Pooling	2x2	–	126x94x32
Convolution 3	3x3	96	124x92x48
Pooling	2x2	–	62x46x48
Convolution 4	3x3	128	60x44x64
Pooling	2x2	–	30x22x64
Convolution 5	3x3	128	28x20x80
Pooling	2x2	–	14x10x80
Convolution 6	3x3	192	12x8x96
Pooling	2x2	–	6x4x96
Convolution 7	3x3	256	4x2x96
Pooling	2x2	–	2x1x96
Fully Connected 1	–	512	128
Fully Connected 2	–	512	128
Fully Connected 3	–	256	256
Fully Connected 4	–	5	5

## MODEL PERFORMANCE DETAILS

Tables B and B detail the performance of all predictive models on artificial and real images respectively. Measures shown are Expected error per bag ( $e$ ) followed by standard deviation of error per bag ( $\rho$ ) in parentheses:  $e(\rho)$ . Note that the Linear Regression, SVR and CNN use a median over 33 independent simulations of a set definition as the prediction for that set of images except in the case of Real images where a sequence of 33 was not available. Aggregating predictions over 33 examples is irrelevant for Guess the mean and Single bag sampling baseline methods. Table B shows the outputs selected for each training context as well as the image set sizes. The values in bold represent the best performance for the respective output and data set. A \* indicates significantly different prediction error than mean guess at  $\alpha = 0.05$ , and \*\* indicates significantly different prediction error than mean guess at  $\alpha = 0.01$ , and are only shown for those measures in bold. Values in the Non Realism data set column with a † are performance on the Realism outputs described in Table B

**Table B.1: The types of outputs and set sizes for each training context.**

	No Realism		Realism	
	Historical	Mixed	Historical	Mixed
Network Outputs	% < 2" (as ratio)		% < 2" (as ratio)	
	% < 3" (as ratio)		% 2" – 3" (as ratio)	
	% < 4" (as ratio)		% 3" – 4" (as ratio)	
	% > 4" (as ratio)		% > 4" (as ratio)	
	Mean Length		Mean Length	
Training Set Size	13002	10026	7125	21285
Validation Set Size	3267	2540	726	5346
Test Set Size	305			

**Table B.2: Performance of models against artificial test images by the type of data they were trained on. See the top of this section for a description.**

Method	No Realism		Realism	
	Historical	Mixed	Historical	Mixed
Guess The Mean	0.032(0.027)	0.115(0.086)	<b>0.027(0.014)</b>	0.270(0.146)
	0.036(0.031)	0.151(0.120)	<b>0.027(0.020)</b>	0.194(0.142)
	0.021(0.017)	0.101(0.075)	0.027(0.018)	0.164(0.136)
	0.021(0.017)	0.141(0.109)	0.022(0.014)	0.158(0.138)
	0.088(0.083)	0.333(0.258)	0.090(0.063)	0.682(0.509)
1 Bag Sampling	0.046(0.037)	0.163(0.121)	0.037(0.027)	0.350(0.257)
	0.052(0.043)	0.151(0.120)	0.038(0.028)	0.245(0.236)
	0.030(0.023)	0.141(0.109)	0.037(0.028)	0.218(0.209)
	0.030(0.023)	0.141(0.109)	0.030(0.022)	0.209(0.210)
	0.131(0.111)	0.469(0.367)	0.125(0.063)	0.963" (0.723)
Linear Regression	0.028(0.024)	0.086(0.069)	0.029(0.016)	0.128(0.092))
	<b>0.029(0.024)*</b>	0.061(0.057)	0.028(0.021)	0.159(0.128)
	<b>0.016(0.012)</b>	0.063(0.044)	0.030(0.018)	0.136(0.123)
	0.028(0.024)	0.063(0.044)	<b>0.020(0.012)</b>	0.074(0.071)
	<b>0.029(0.024)*</b>	0.159(0.134)	<b>0.073(0.060)</b>	0.154(0.117)
SVR	0.032(0.027)	0.124(0.084)	0.028(0.019)	0.266(0.163)
	0.036(0.031)	0.113(0.106)	0.029(0.020)	0.205(0.131)
	0.022(0.016)	0.106(0.074)	<b>0.027(0.016)</b>	0.170(0.132)
	0.022(0.016)	0.106(0.074)	0.024(0.013)	0.157(0.138)
	0.089(0.082)	0.368(0.271)	0.087(0.064)	0.679(0.523)
CNN	<b>0.028(0.022)</b>	<b>0.056(0.040)**</b>	0.035(0.025)	<b>0.095(0.084)**</b>
	0.026(0.020)†	<b>0.049(0.036)**</b>	0.031(0.025)	<b>0.126(0.130)**</b>
	0.031(0.021)†	<b>0.045(0.036)**</b>	0.035(0.025)	<b>0.121(0.115)*</b>
	<b>0.017(0.013)*</b>	<b>0.044(0.036)**</b>	0.026(0.021)	<b>0.062(0.063)**</b>
	0.072(0.055)	<b>0.057(0.054)**</b>	0.125(0.096)	<b>0.131(0.096)**</b>

**Table B.3: Performance of models against real test images by the type of data they were trained on. See the top of this section for a description.**

Method	No Realism		Realism	
	Historical	Mixed	Historical	Mixed
Guess The Mean	<b>0.036(0.027)</b> <b>0.047(0.037)</b> <b>0.028(0.023)</b> <b>0.028(0.023)</b> <b>0.119(0.098)</b>		<b>0.036(0.027)</b> <b>0.029(0.022)</b> <b>0.026(0.018)</b> 0.028(0.032) <b>0.119(0.098)</b>	
by 1 Bag Sampling	0.050(0.039) 0.067(0.051) 0.040(0.032) 0.040(0.032) 0.172(0.135)		0.050(0.039) 0.042(0.031) 0.036(0.027) 0.040(0.032) 0.172(0.135)	
Linear Regression	0.412(0.694) 0.585(0.488) 0.404(0.402) 0.404(0.402) 1.572(1.388)	0.699(3.603) 0.329(0.653) 0.765(1.925) 0.765(1.925) 1.181(4.722)	0.145(0.676) 0.127(0.394) 0.086(0.226) 0.136(0.420) 0.495(1.803)	0.189(0.151) 0.193(0.163) 0.185(0.180) 0.155(0.160) 0.521(0.503)
SVR	0.038(0.031) 0.047(0.037) 0.028(0.024) 0.028(0.024) 0.126(0.100)	0.172(0.045) 0.053(0.041) 0.045(0.027) 0.045(0.027) 0.318(0.147)	0.037(0.031) 0.035(0.026) 0.026(0.019) <b>0.028(0.024)</b> 0.124(0.099)	0.138(0.044) 0.125(0.037) 0.044(0.027) 0.045(0.028) 0.254(0.138)
CNN	0.038(0.031) 0.031(0.023)† 0.027(0.019)† 0.030(0.026) 0.134(0.110)	0.106(0.065) 0.130(0.084) 0.254(0.093) 0.260(0.102) 0.284(0.274)	0.066(0.043) 0.051(0.037) 0.052(0.035) 0.070(0.061) 0.329(0.203)	0.169(0.127) 0.270(0.107) 0.187(0.109) 0.272(0.227) 0.523(0.474)

# ARTIFACTS

Exploration and validation of this thesis has produced several artifacts.

First of these is a set of software used for creating virtual radiation tomography images of bagged product. This is in the form of a pipeline of 4 separate software. First is a program and associated data files used to generate sets of product statistics. Second is a set of scripts to be run in the Blender [2] software environment capable of generating realistic stochastic meshes. Third is a software utilizing the Bullet physics simulation library [4] capable of reading a set of meshes, a bag model, and then simulating a bagging process. Finally, a software utilizing the gVirtualX-ray [8] library capable of generating virtual tomographic images of a set of models output by the physics simulator is realized. As a sequence these software create artificial exposures of bagged product.

Additionally a set of software tools used to extract image features and subsequently train a Convolutional Neural Network are produced. Facilities to use the resulting trained CNN on new images is also created.

