Spring 5-2022

Deep Learning Object-Based Detection of Manufacturing Defects in X-ray Inspection Imaging

Juan C. Parducci
Old Dominion University, juan.parducci@gmail.com

Deep Learning Object-Based Detection of Manufacturing Defects in X-ray Inspection Imaging

by

Juan C. Parducci
B.S. May 2012, Old Dominion University

A Thesis Submitted to the Faculty of
Old Dominion University in Partial Fulfillment of the
Requirements for the Degree of

MASTER OF SCIENCE

AEROSPACE ENGINEERING

OLD DOMINION UNIVERSITY
May 2022

Approved by:

Shizhi Qian (Director)

Xiaoyu Zhang (Member)

Yan Peng (Member)

ABSTRACT

DEEP LEARNING OBJECT-BASED DETECTION OF MANUFACTURING DEFECTS IN X-RAY INSPECTION IMAGING

Juan C. Parducci
Old Dominion University, 2022
Director: Dr. Shizhi Qian

Current analysis of manufacturing defects in the production of rims and tires via x-ray inspection at an industry partner's manufacturing plant requires that a quality control specialist visually inspect radiographic images for defects of varying sizes. For each sample, twelve radiographs are taken within 35 seconds. Some defects are very small in size and difficult to see (e.g., pinholes) whereas others are large and easily identifiable. Implementing this quality control practice across all products in its human-effort driven state is not feasible given the time constraint present for analysis.

This study aims to identify and develop an object detector capable of conducting defect detection in real-time across all manufactured products to remove the human-in-the-loop from the quality control cycle and leverage subject matter expertise at scale.

A survey of existing literature on object detectors in a defect detection setting was conducted to aid in the selection process of the detector algorithm. Of focus were studies related to the inspection of radiographs in real-time. Additional consideration was given to studies where the performance of small object detection was characterized. Following the literature review, defect detection models were trained and assessed for performance. The object detector utilized in this study is YOLOv3 with a Darknet-53 network.

The first trained model, where all twelve defect classes were considered, had the lowest performance metrics across precision, recall, F1-score, and accuracy of 70.9%, 74.7%, 72.8%, and 70.7% respectively at an IoU threshold of greater than zero. The highest performing model combined defects belonging to defect sub-classes into a super class to reduce model ambiguity. The final model had a precision, recall, F1-score, and accuracy of 92.2%, 92.3%, 92.2%, and 92.5% respectively.

The final YOLOv3 model performed significantly better than the model trained with the data as originally provided through the application of data relabeling. In addition to selecting the appropriate object detector and network in the development of a machine learning model, it is important to have standardized defect annotation guidelines in place to produce consistent data labeling results. Lastly, the selection of sufficiently distinct defect classes for detection is vital for optimal results.

To my wife, Jessica, thank you for your love and constant support. Thank you for your patience and unwavering belief in me. Most importantly, thank you for being my best friend.

A special thanks to Rebecca for her continued support. Thank you for making the world a better place.

ACKNOWLEDGMENTS

I would like to extend my gratitude to my advisor, Dr. Qian, for his continuous support throughout the development of this thesis. His constant availability, feedback, and encouragement have made this possible. I would also like to extend my thanks to my committee members for their guidance.

I would also like to extend my gratitude to the industry partner who provided the raw data used in this study.

NOMENCLATURE

ANN	Artificial Neural Network
CNN	Convolutional Neural Network
COCO	Common Objects in Context
FCN	Fully Convolutional Network
FN	False Negatives
FP	False Positives
IoU	Intersection Over Union
mAP	Mean Average Precision
R-CNN	Region-based Convolutional Neural Network
RPN	Region Proposal Network
SGD	Stochastic Gradient Descent
SSD	Single Shot MultiBox Detector
SUN	Scene Understanding
TN	True Negatives
TP	True Positives
VGG	Visual Geometry Group
YOLO	You Only Look Once

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

CHAPTER 1

INTRODUCTION

Object detection is a technique in computer vision capable of both automated classification and localization tasks through the use of deep learning, a machine learning technique within the field of artificial intelligence (Choi et al., 2020). Industrial applications of object detection range from autonomous vehicle navigation, medical imaging diagnostics, identity management, quality control defect detection, to image and video tracking (Vahab et al., 2019). The automation offered by these technologies enables businesses to improve performance in task completion time while improving quality and reducing error rates. These technologies ultimately achieve outcomes that extend beyond human capabilities (McKinsey & Company, 2017).

In this study, the application of deep learning techniques is applied to defect detection in the manufacturing setting of tires and rims with the objective serving as an automated quality control measure. An object detector is trained, through supervised learning, to perform the task of defect detection by learning the visual criteria by which a human determines whether a product is defective or defect-free.

This thesis is part of a collaborative effort between Old Dominion University and an industry partner. Old Dominion University was provided with the raw ground truth data utilized in this study.

1.1 Conceptual Framework

The conceptual framework adopted for the development, assessment, and iteration cycles of the deep learning models in this study is based off of the IBM® (2017) machine-learning

model creation workflow. The portions of this workflow in Figure 1.1 related to data acquisition have been omitted as Old Dominion University was provided the raw ground truth data.

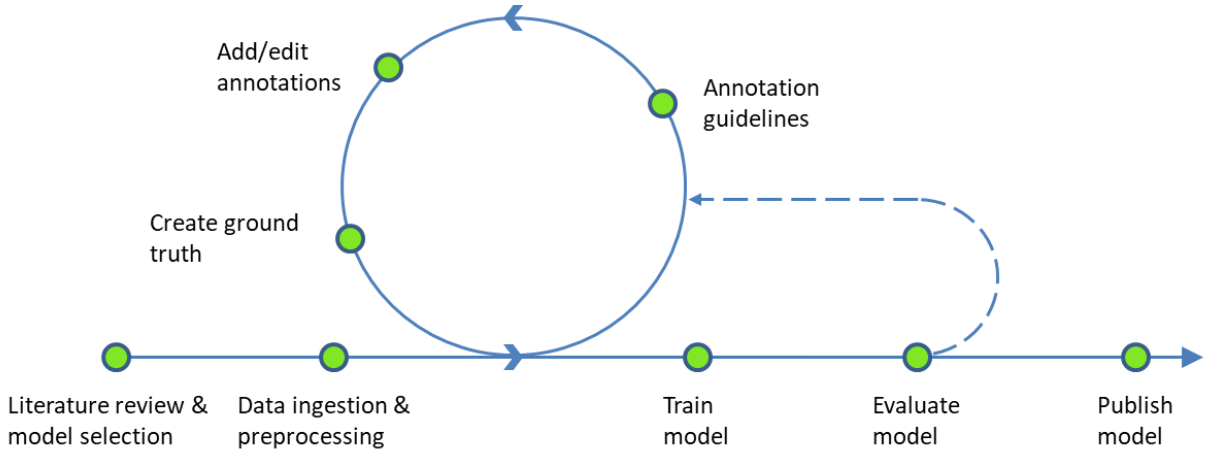

Figure 1.1: Conceptual Framework

1.2 Purpose

The aim of this study is to develop an object detector capable of detecting manufacturing defects in tires and rims in real-time and remove defective products from production. In its current state, quality control is being conducted through human efforts where a quality control specialist manually reviews radiograph of selected samples for defects. If the specialists were to review samples in real-time, they would be afforded approximately 35 seconds to assess 12 radiographs for defects, some of which are very small and difficult to identify at first glance. Given the time constraints in a constant production environment, human effort is not a viable solution for quality control at scale.

The subject matter expertise from these quality control specialists is to be leveraged through the development of a machine learning model capable of making the same observations and following the decision-making process that a specialist would across all manufactured

products in real-time. In addition to serving as a quality control method, additional insight into the types of defects that are present in the final products will aid the quality assurance process. There are twelve defect classes of interest in the data provided to Old Dominion University. A description of each defect class and some of their possible manufacturing causes are described below.

1. Burr – a common defect in shearing and cutting operations (Rai et al., 2013) where undesirable projections of the material are formed. A general definition of a burr extends beyond cutting and shearing and is defined as an imperfection created during the manufacturing process which results in a body of material which extends beyond the work piece (Aurich et al., 2009). Figure 1.2 shows a sample burr defect on a tire wall from the molding process.

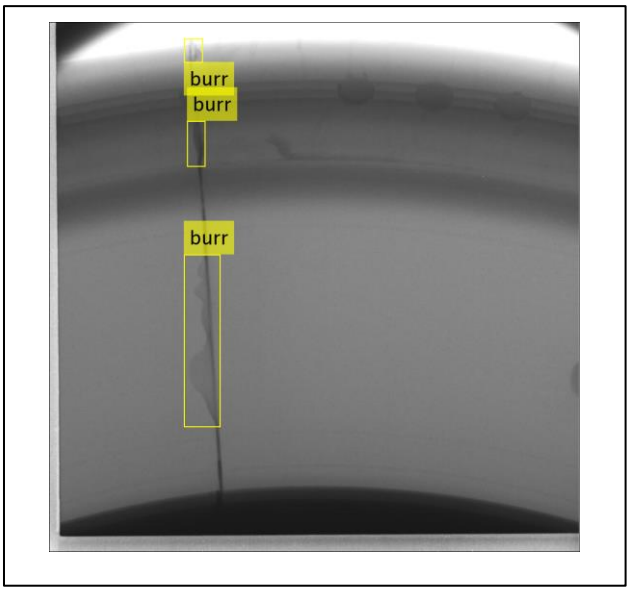

Figure 1.2: Burr

2. Cavity – appears as spherical/rounded voids which can be divided into pin holes, or larger blowholes. Causes include entrapment of air during liquid metal pouring, high moisture content, and inadequate vents (Juriani, 2015). Figure 1.3 shows a cavity in a wheel spoke and Figure 1.4 shows small cavities on a tire wall.

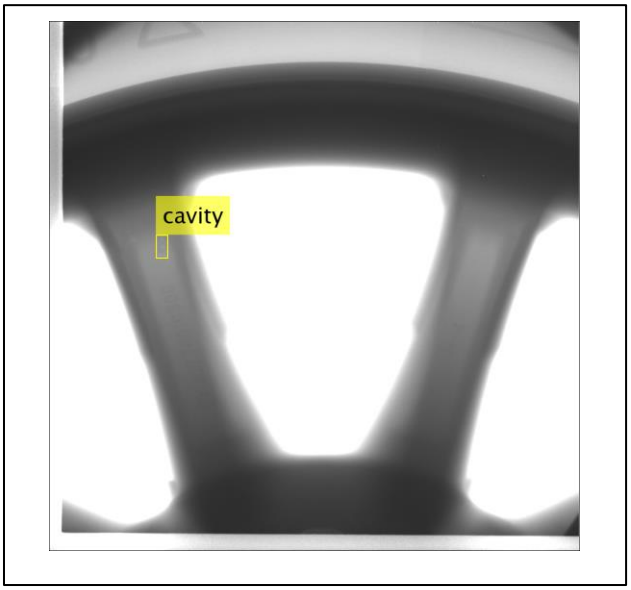

Figure 1.3: Cavity in Wheel Spoke

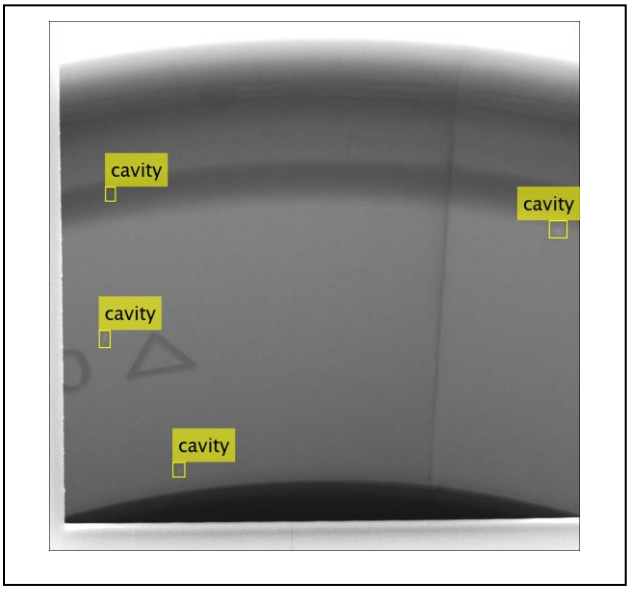

Figure 1.4: Cavity in Tire Wall

3. Crackle – a grouping of cracks. Some causes are resulting from rapid and non-uniform cooling of metals, shrinking of casting, and imperfections in the mold (Ingle, 2017). Figure 1.5 shows crackle formation on a wheel.

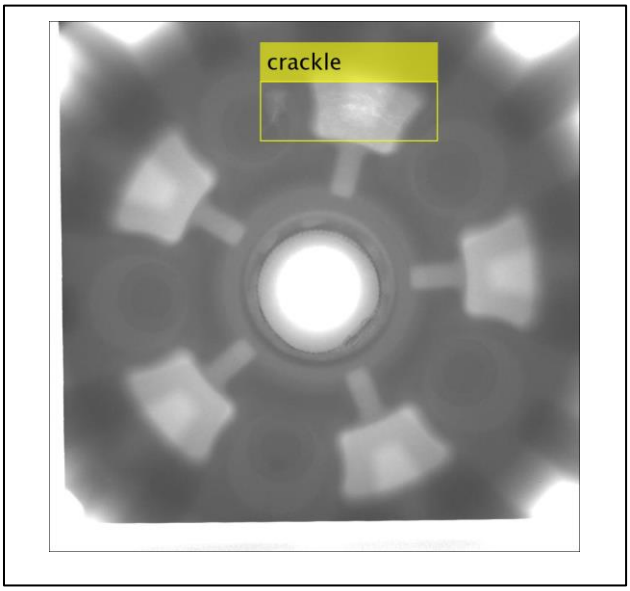

Figure 1.5: Crackle

4. Curl dregs – slags found on the material surface and are caused when molten metal containing slag is poured into a mold. This type of defect can be remedied by removing slag particles from the molten metal prior to pouring via filters or adding additives to the molten metal mixture to allow slags to float to the top for removal (Wong, 2018). Figure 1.6 shows the deposition on curl dregs on a wheel hub.

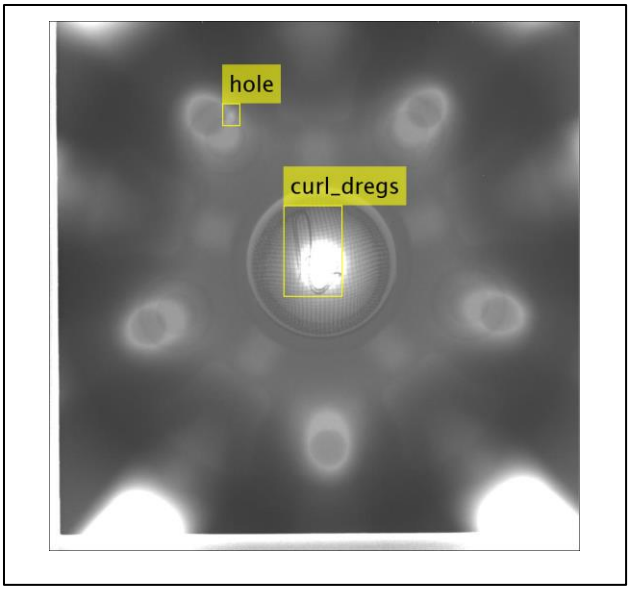

Figure: 1.6: Curl Dregs

5. Dregs – a shorter variant of the curl dregs defect. Figure 1.7 shows the deposition of dregs on a wheel hub.

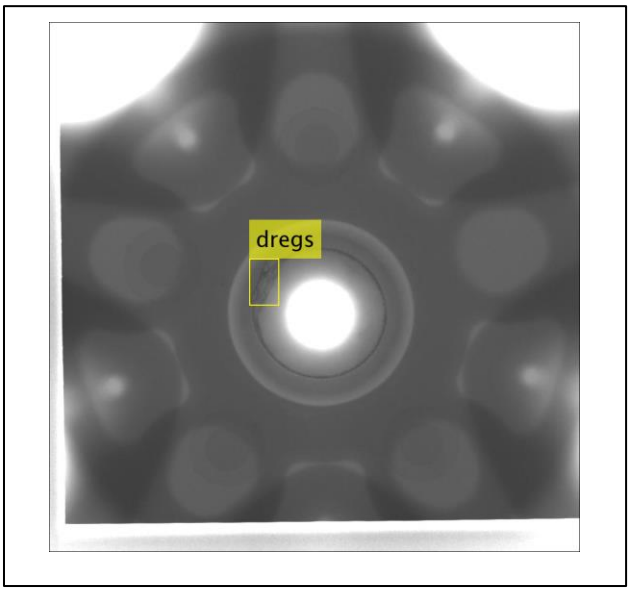

Figure 1.7: Dregs

6. Hole – a variant of the cavity defect. Figure 1.8 shows holes at lug nut holes.

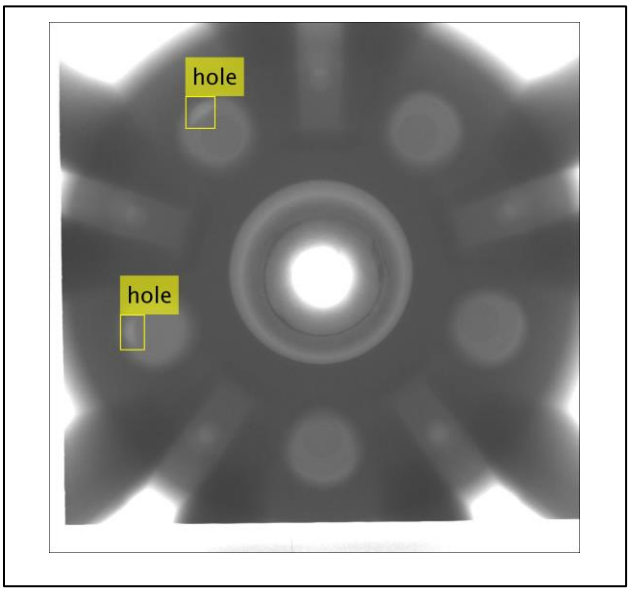

Figure 1.8: Hole

7. Incomplete – a defect where the tire did not fully form in the mold. Shown in Figure 1.9.

Figure 1.9: Incomplete in Tire

8. Long burr – a variant of the burr defect as shown in Figure 1.10.

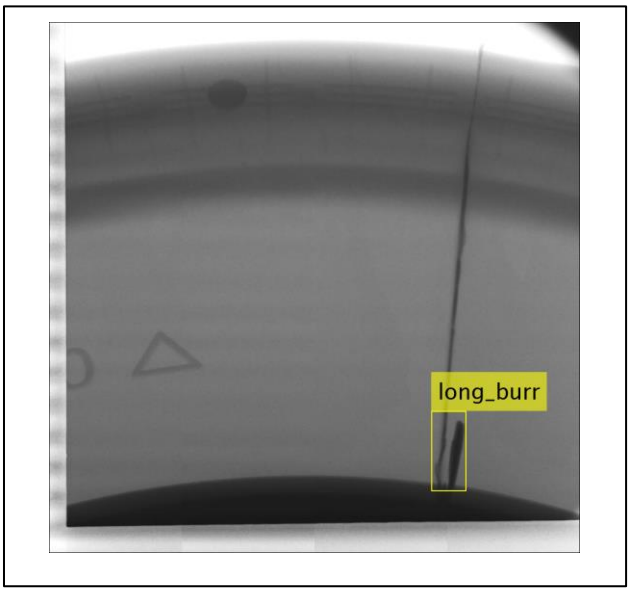

Figure 1.10: Long Burr

9. Low dregs – a variant of the curl dregs defect. Dregs in the tire can be caused by the use of a dirty mold during vulcanization (Weyssenhoff et al., 2019). See Figure 1.11.

Figure 1.11: Low Dregs

10. Shrinkage – a cavity caused as the cast wheel cools due to poor metal volume, uneven metal distribution, or contaminated metal. (Patil et al., 2015). See Figure 1.12.

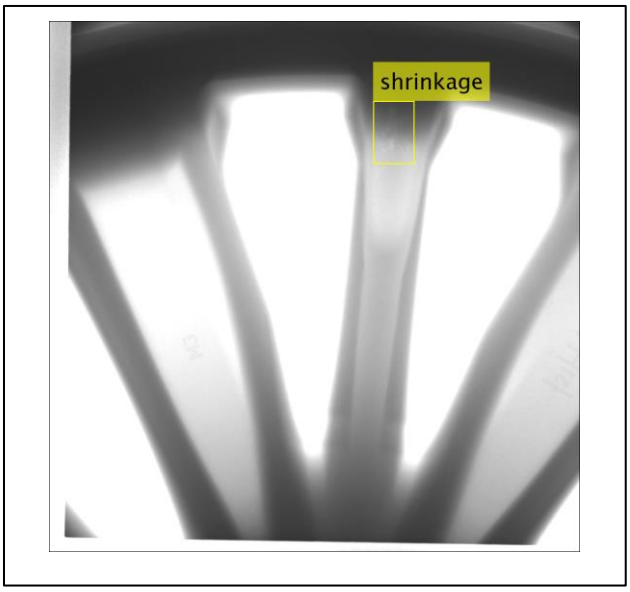

Figure 1.12: Shrinkage

11. Side – a variation of the cavity defect related to the tire only. See Figure 1.13.

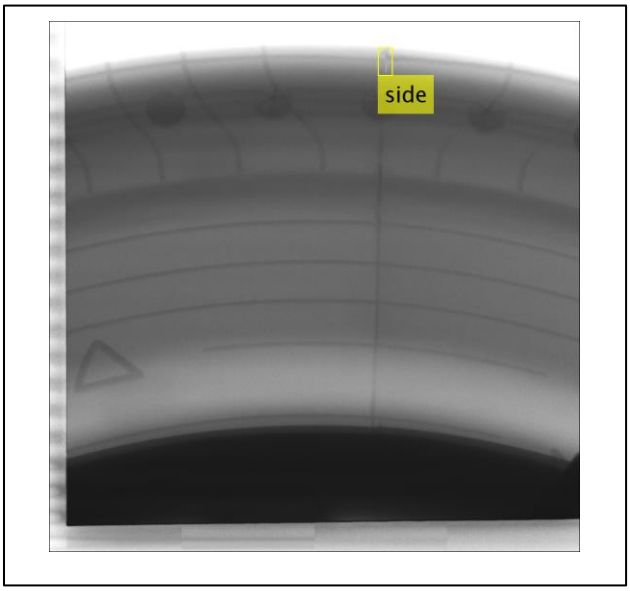

Figure 1.13: Side

12. Stripe incomplete – a larger variant of the cavity defect. See Figure 1.14.

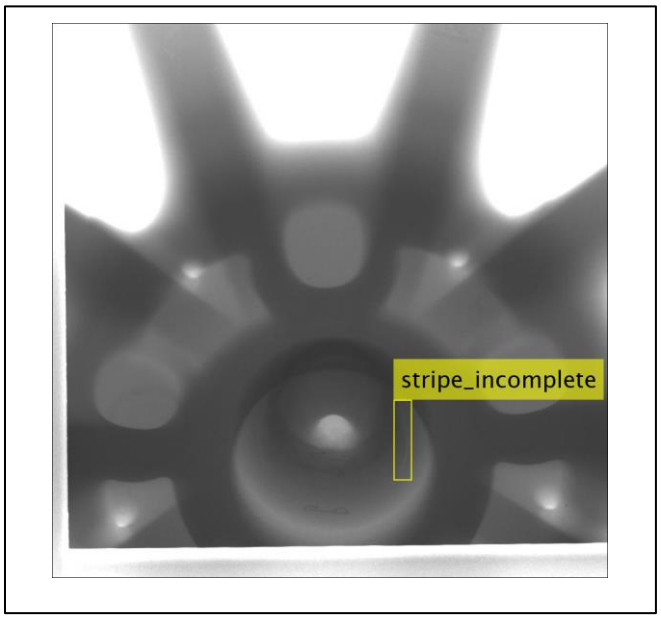

Figure 1.14: Stripe Incomplete

1.3 Problem

There are challenges present from the data preparation stage of the project such as a lack of consistent data labeling practices and general considerations when selecting defects to be identified in terms of their uniqueness. The detection of small objects presents its own challenge.

1.3.1 Data Annotations

When it comes to machine learning projects, having standardized defect definitions and labeling practices establishes a reliable set of ground truth data which is one of the most important requirements of a successful project (Zhou, 2021). A review of the data supplied to Old Dominion University shows that there are discrepancies in the data labeling practices used by the manufacturer. For instance, in some cases where there are many defects clustered together, one data labeler may have labeled each occurrence individually while another labeler

might have drawn one single bounding box around all the defects present. Another related issue is the failure to identify defects by the data labelers. One example of this can be seen in Figure 1.15, where there are three dreg defects that were not labeled by the subject matter experts. The missed defects have been highlighted in red.

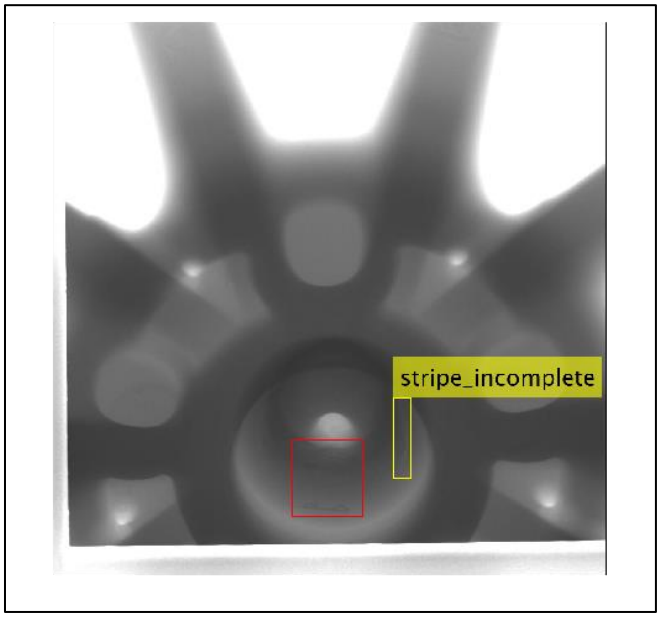

Figure 1.15: Missed Defect Annotation

Given the defect labels present in the dataset along with their visual characteristics, available in section 1.2, another potential labeling problem is one where some of the defects are not sufficiently different from one another. For example, the defects labeled as cavity, crackle, hole, low dregs, and shrinkage all appear very similar in terms of geometry and brightness in the radiographs when compared to the sample background. This could cause poor detector performance between these classes.

Additionally, the defects labeled as curl dregs, incomplete, long burr, low dregs, side, and stripe are relatively rare in occurrence when compared to the rest of the data. This causes poor model representation for these defects due to the greater probability of a predicted defect class

belonging to one of the more represented classes (Rocca, 2019). A summary of the defect distribution present in the data is available in Figure 1.16. Although there is data imbalance present in this study, it is representative of the defects encountered in the production environment.

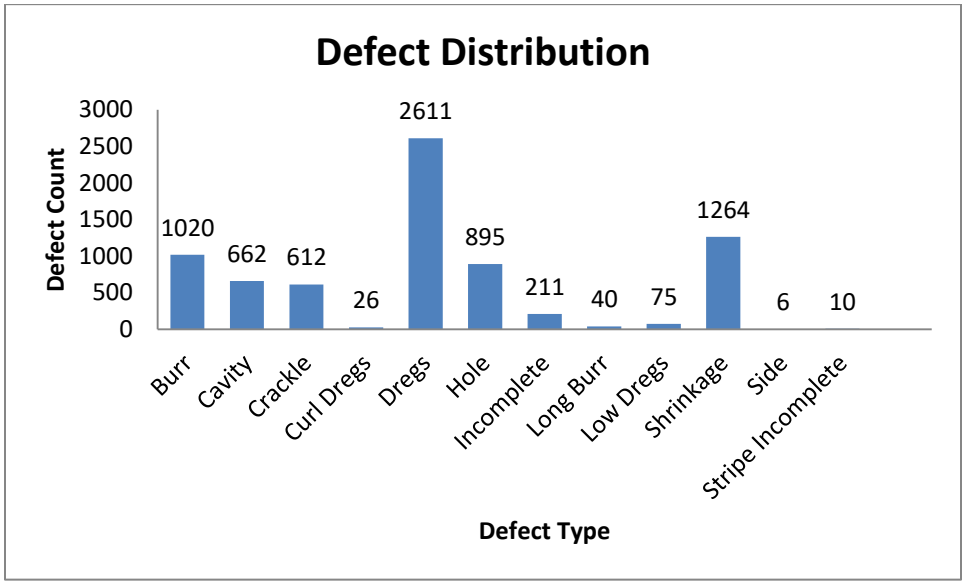

Figure 1.16: Defect Distribution

1.3.2 Small Object Detection

There is an inherent challenge in the detection of small objects. In general, object detectors have a hard time detecting small objects in comparison to large objects. The mean average precision for large object detection compared to small object detection may be anywhere between two to five times as high depending on the detector and network combination (Solawetz, 2020).

1.4 Method and Procedure

The data provided to Old Dominion University is in the form of images and corresponding xml files. In order to train the object detector, the data will need to undergo a data ingestion process to build the necessary data structures and image repositories to make the data usable in the software environment. The data will then be divided into a training dataset and test dataset in a 70% and 30% split respectively. The division will be based on defect classes, not number of images, as some images contain more than one defect, and some defects are rarer than others.

The selection of the object detector and the detector network to be utilized will be based primarily on the results of past works related to defect detection in radiographs. Additional consideration will be given to the object detectors capable of real-time performance and their ability to detect small objects.

During model training, data augmentation techniques will be applied at random within a specified threshold to synthesize additional data to prevent model overfitting (Ying, 2019) and maintain good generalization. As stated in the problem section of chapter one, there are concerns regarding the defect labeling convention used by the manufacturer where a few of the defect classes appear to be subclasses of the cavity defect. Analysis of models will take this into consideration.

The evaluation of the machine learning models and their iterations will follow the workflows depicted in section 1.1 and section 3.4.2.

CHAPTER 2

BACKGROUND OF THE STUDY

This chapter defines the concept of radiographic defect detection as a method for quality control in a manufacturing setting. A background in machine learning has been provided in order to further explore the application of machine learning methods in the area of object detection. The detector of interest in this review is the You Only Look Once (YOLO) family of detectors, as it is the detector utilized in this study. Related works are presented at the end of the chapter.

2.1 Radiographic Defect Detection

Part of the quality control process requires that manufactured products be inspected for deviations from a set standard prior to acceptance. In this study, one non-destructive method of inspection being utilized is radiographic imaging in the detection of defects. X-ray images are 2D projections of 3D objects capable of revealing hidden features within the object (Ren et al., 2019). Through the use of radiographs, defects such as crackles, holes, and cavities can be observed. The discovery of internal defects in industrial radiography is of importance, especially as the concept of industrial automation continues to evolve and replace human effort with efficient systems (Fadel et al., 2020).

2.2 Machine Learning

Machine learning is a technique within the field of artificial intelligence. Machine learning differs from traditional programming in the sense that a traditional program requires the creation of detailed instructions for a computer to follow. In some cases, creating a traditional program can be time exhausting or near-impossible, specifically when training a computer to

recognize images or features/objects within an image. While humans can accomplish this task with relative ease, it is difficult to write explicit instructions for a computer to perform this task. Machine learning takes the approach of supplying computers with a set of data, called training data (e.g., images, numerical data), and enables computers to learn to program themselves through experience by finding patterns. The result is a machine learning model which can make predictions when provided with a different set of data (Massachusetts Institute of Technology, 2021).

2.2.1 Types of Machine Learning

Machine learning algorithms are generally divided into four categories, each with its own suitability in solving problems (Sarker, 2021).

- Supervised learning – the type of learning conducted in this study, uses a collection of training data, with labeled data, to infer a function (i.e., task-driven approach). The most common supervised tasks are those of classification and regression. In this study, for example, supervised learning is being utilized to locate defects within an image and provide defect classification.
- Unsupervised learning – utilizes analyzed unlabeled data without the need for human interference (i.e., data-driven process). This type of learning is generally used for identifying meaningful trends, clustering in results, and exploratory purposes.
- Semi-supervised learning – can be defined as a hybrid (i.e,. hybridization) between supervised and unsupervised learning methods as it utilizes both labeled and unlabeled data (Sarker, 2021, as cited in Han et al., 2011, and Sarker et al., 2020). Some

applications of semi-supervised learning include machine translation, fraud detection, labeling data, and text classification.

- Reinforcement learning – enables software and machines to automatically evaluate optimal behavior in order to improve its efficiency (i.e., environment-driven approach) (Sarker, 2021, as cited in Kaelbling et al., 1996). It is a useful method for training models where optimization is desired (e.g., operational efficiency and autonomous tasks).

2.2.2 Deep Learning

Deep learning is a machine learning technique that mimics the way the human brain processes data. Deep learning models are trained by using large datasets and neural networks with multiple processing layers that learn representations of data with many levels of abstraction. When it comes to performing object detection via deep supervised learning, a large set of labeled data (i.e., ground truth dataset) is processed through artificial neural networks (ANNs) whose output is a vector of scores representing the probability distribution of each potential outcome (Alzubaidi et al., 2021).

In machine learning, backpropagation is an algorithm used to determine how a machine should change its variables (i.e., weights) which are used in the representation of each of these processing layers. The machine will iteratively update the weights of the algorithm as it makes multiple passes through the dataset by calculating a gradient vector indicating an increase or decrease in detection error given a small change in the weights. The weight vector is then adjusted inversely to the gradient vector in a procedure called stochastic gradient descent (SGD), where the error is brought closer to a minimum. When it comes to the application of supervised learning, this is the procedure most utilized by practitioners (LeCun et al., 2015).

2.2.3 Artificial Neural Networks

Artificial neural networks (ANN) share the concept of neurons in their architecture where each neuron can be interpreted as a human neuron in biology. Each artificial neuron is composed of activation functions that control the propagation of an artificial neuronal signal to the next layer where a positive weight provides excitatory stimulus and negative weights provide inhibitory ones (Suzuki, 2013). In machine learning, a synapse is the connection between inputs to neurons, neurons to neurons and neurons to outputs. Each connection has a unique synapse with unique weights. When discussing the adjustment of weights, it is the weight of these synapses that is being referenced (Deep AI, 2019). Figure 2.1 depicts an artificial neuron and synapses.

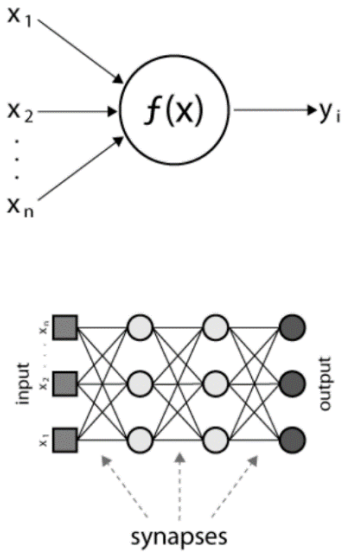

Figure: 2.1: Artificial Neural Network

Note. Adapted from "Artificial Neural Network" by K. Suzuki, 2013, London, United Kingdom: IntechOpen. CC License.

2.2.4 Convolutional Neural Networks

Convolutional neural networks (CNN) are a class of artificial neural networks that are prevalent in computer vision tasks. CNNs are designed to automatically and adaptively learn features through backpropagation by using multiple layers including convolution layers, pooling layers, and fully connected layers (Yamashita et al., 2018).

- Convolution layer – performs feature extraction through mathematical operations resulting in a feature map representing different characteristics. Convolution involves taking a small array of numbers called a kernel, or filter, and calculates an element-wise product between the kernel and the input tensor to generate a feature map. This process is depicted in Figure 2.2.
- Pooling layer – provides downsampling of the feature map generated from the convolutional layer.
- Fully connected layer – provides mapping of the downsampled feature maps to the final network outputs (e.g., probabilities in classification tasks)

Although there are many CNNs, the focus will be on the YOLO family of detectors as it is the CNN utilized in this study.

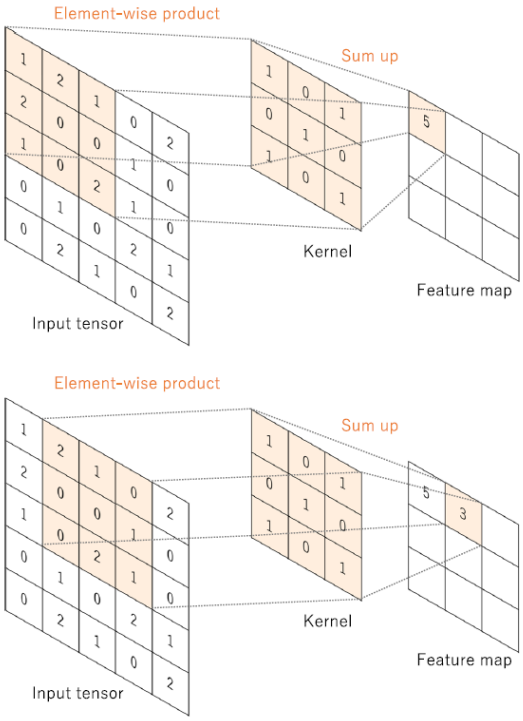

Figure 2.2: Convolutional Operation

Note. Adapted from "Convolutional neural networks: an overview and application in radiology" by R. Yamashita, M. Nishio, R. K. G. Do, and K. Togashi, 2018, *Insights into Imaging, 9*, p. 614. (https://doi.org/10.1007/s13244-018-0639-9) CC License.

2.3 Object Recognition and Detection

Object recognition is a computer vision technique within the field of machine learning and artificial intelligence which aims to identify objects of interest within an image or video. Some industrial applications include target tracking, autonomous vehicle navigation, surveillance, medical image analysis, and industrial inspection (Deng et al., 2020). Object recognition has two parts:

- Image classification – an algorithm takes an image as an input and assigns a classification label (e.g., car, person).

- Object localization – an algorithm will find an object within a given image and map a bounding box around the object of interest.

Object detection is a hybrid of both image classification and object localization. Object detection takes an input image and is capable of localizing one or many objects of the same or distinct classes and assigns each of them a classification label, a bounding box, and a confidence score representing the likelihood that the bounding box contains an object (Ralasic, 2021). These differences are depicted in Figure 2.3.

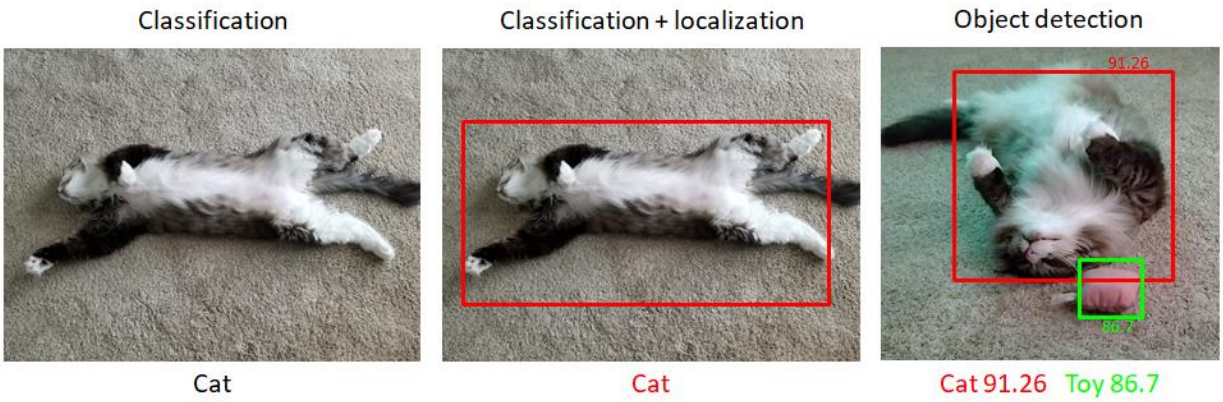

Figure 2.3: Object Detection

Object detection algorithms based on deep learning models are generally divided into two categories and have the following characteristics according to Soviany and Ionescu (2018):

- Single-stage detectors – such as Single Shot MultiBox Detector (SSD) and YOLO take an input image or video and learn the class probabilities and bounding box coordinates via regression. These models are not as accurate as two-stage detectors but perform detection tasks much faster.

- Two-stage detectors – such as Region-based Convolutional Neural Network (Faster R-CNN) use a Region Proposal Network (RPN) to generate regions of interest in the first stage, and in the second stage perform object classification and bounding box regression. These models achieve the highest accuracy but are typically slower.

2.4 You Only Look Once (YOLO)

YOLO, first published in 2015, uses a single CNN to predict multiple bounding boxes and object class probabilities from images in one pass by treating object detection as a regression problem. Figure 2.4 shows a depiction of this process. During training, and when making predictions, YOLO sees the entire image and encodes contextual information about classes and their features. It is because of the ability to see the entire image that YOLO makes less background errors compared to a region proposal-based detector like Fast R-CNN. Fast R-CNN mistakes background features during object detection due to its inability to see the "big picture" (Redmon et al., 2016).

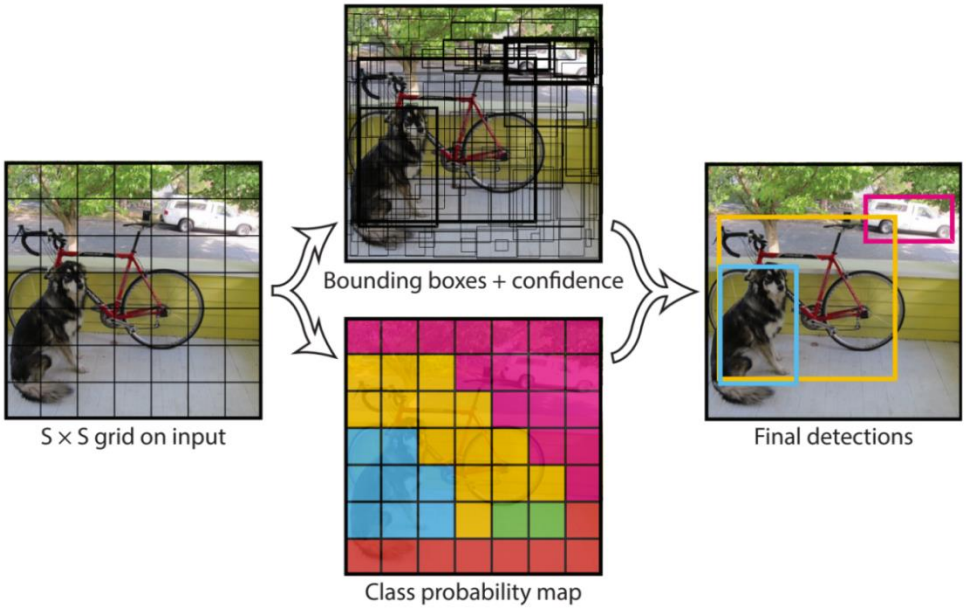

Figure 2.4: Class Probability and Bounding Box Predictions

Note. From "You Only Look Once: Unified, Real-Time Object Detection" by J. Redmon, S. Divvala, R. Girshik, and A. Farhadi, 2016, *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, p. 780. (https://doi.org/10.1109/CVPR.2016.91).

Despite its benefits, YOLO's accuracy is less than that of other state-of-the-art detectors. Each grid square can only predict two bounding boxes and can have an object hindering the number of nearby objects YOLO can predict. It also struggles with small object detection (Redmon et al., 2016). These shortcomings have been addressed in subsequent iterations of YOLO which will be further discussed.

2.4.1 YOLOv2

Also known as YOLO9000, YOLOv2 includes several improvements to the original version which have increased its mean average precision (mAP) on the Pascal VOC2007 dataset

from a mAP of 63.4% to 78.6%. The network utilized for feature extraction is a model called Darknet-19 which has 19 convolutional layers and 5 maxpooling layers. A few of these improvements according to Redmon and Farhadi (2017) include:

- Anchor boxes – in this version, the fully connected layers from YOLOv1 have been removed and the prediction of bounding boxes is achieved through the use of anchor boxes. Anchor boxes are a set of user-defined bounding boxes, through a practice called k-means clustering, which captures the aspect ratio of object classes most likely to produce quality detections. The aspect ratio of these boxes is determined from the training dataset.
- Multi-scale training – YOLOv2 consists of convolutional and pooling layers that enable it to resize the input image size on demand.
- High resolution classifier – The original YOLO trained networks at a resolution of 224x224 pixels. It has been increased to 448x448 pixels.
- Other improvements include batch normalization to help remove model dropout without overfitting and the addition of a pass-through layer for localizing smaller objects.

2.4.2 YOLOv3

YOLOv3 includes minimal but important updates to YOLOv2 which makes it more capable of detecting small objects. The major improvements according to Redmon and Farhadi (2018) include:

- Feature extractor – this new version utilizes a new feature extraction network called Darknet-53 which is composed of 53 convolutional layers for enhanced feature extraction. It is much more robust than Darknet-19 and larger. Yet it is much more

efficient than two other object detector backbones that contain significantly more layers: ResNet-101 and ResNet152.

- Multi-scale prediction – YOLOv3 predicts bounding boxes across three different scales. Three bounding boxes are predicted at each of the three scales and continue to use k-means clustering to determine bounding boxes.

2.5 Related Work

This study focuses on the detection of small objects in radiographs, specifically manufacturing defects. There are few studies available that directly compare various object detectors (e.g., SSD, YOLO, Faster R-CNN) against each other when tasked with defect detection in x-ray images. In addition to highlighting works related to tire defect detection, additional studies involving object detection within radiographs, and small object detection studies will be reviewed.

Liu et al. (2015) have proposed a weld defect detection method via x-ray imaging based on YOLOv3 called LF-YOLO. They have compared the performance of various object detectors including SSD, YOLOv3, Faster R-CNN, and RetinaNet against their proposed LF-YOLO method – where LF stands for Lighter and Faster. Of the single-stage detectors tested, the highest performing model was LF-YOLO (mAP 92.9%) while the second highest mAP was achieved by YOLOv3 (mAP 91.0%) utilizing a Darknet-53 backbone. Of the two-stage detectors, Faster R-CNN performed the best (mAP 92.2%) with a ResNet101 backbone.

Nguyen et al. (2020) conducted an extensive assessment of small object detection by testing various permutations of object detectors with different backbones at varying image resolutions. This was largely motivated by the fact that most state-of-the-art object detectors

struggle with the detection of small objects. The dataset used in the evaluation was a combination between the Microsoft Common Objects in Context (COCO) dataset and the Scene Understanding (SUN) dataset that consist of common small objects (e.g., computer mouse, switch, outlet, jar). The single-stage object detectors of interest in their study included YOLOv3, SSD, and RetinaNet. Two-stage object detectors of interest were R-CNN, Fast R-CNN, and Faster R-CNN. Of the single-stage object detectors, YOLOv3 (mAP 33.1%) with a Darknet-53 backbone at a resolution of 608x608 pixels performed the best followed by RetinaNet (mAP 30%) and SSD (mAP 11.32%). The YOLOv3 model had the best prediction time at 0.027 seconds. The authors also found that as they increased the resolution above 608x608 pixels, the mAP would decrease. Of the two-stage detectors, Faster R-CNN (mAP of 41.2) with a ResNeXT-101-64 -4d-FPN backbone had the best performance with a prediction time of 0.286 seconds.

Wang et al. (2019) have proposed a tire defect detection method based on a fully convolutional network (FCN) for defect detection in x-ray imaging through the use of fusion layers to up-sample feature maps and reduce detail loss. Their network architecture, with fusion layers, was evaluated for accuracy against their corresponding non-modified object detectors. The original VGG16 detector achieved an average accuracy of 74.87% while the modified VGG16 detector with three fusion layers achieved an average accuracy of 78.91%. The tire defects of interest in their study include bubbles in tread, bubbles in sidewall, impurity in sidewall, impurity in tread, overlaps in treads, overlaps in sidewalls, and an “others” catch-all group.

Wu et al. (2020) have proposed a tire defect detection method using Faster R-CNN capable of detecting the following defects: sidewall bubbles, foreign matter in the tread, and

foreign matter in the sidewall. The performance of their proposed method was evaluated against a Faster-RCNN detector as well. The total accuracy and mAP of their model was 95.4% and 95.37% respectively. The accuracy and mAP achieved by the original Faster-RCNN model with data augmentation was 88.9% and 93.82% respectively.

Yao et al. (2020) developed a CNN named Pneumonia Yolo (PYolo), an improved version of YOLOv3 for the analysis of radiographs of the lungs. The work focused on developing an efficient end-to-end CNN capable of replacing binary classification algorithms used to diagnose pneumonia. In this study, the performance of PYolo (mAP 46.84%) was measured against the performance of the following detectors: Faster R-CNN (mAP 43.01%), SSD (mAP 40.40%), and YOLOv3 (mAP 43.40%).

To summarize, in the studies that do not involve radiographs where comparisons across object detectors were available, Faster R-CNN achieved the best performance. In the study by Nguyen et al. (2020) concerning small objects not in radiographs, Faster R-CNN (mAP 41.2%) outperformed YOLOv3 (mAP 33.1%) at the expense of a greater than ten-fold increase in prediction time. In studies involving x-ray imaging where comparisons were made across object detection models, YOLOv3 and Faster R-CNN performed comparably to each other and maintained less than a 1.5% difference in mAP. In one of the two studies where YOLOv3 and Faster R-CNN were compared, YOLOv3 performed better than Faster R-CNN and vice versa for the other. Given the available information, this study will focus on utilizing YOLOv3 with a Darknet-53 backbone for the real-time detection of manufacturing defects in radiographs.

CHAPTER 3

METHODOLOGY

The general methodology used in the data preparation, data preprocessing, model configuration and performance evaluation of the YOLOv3 object detectors has been adopted from MathWorks (2022). A summary of this methodology is provided in Figure 3.1. The metrics utilized to assess model performance are further discussed in this chapter along with metrics improvement workflows and techniques.

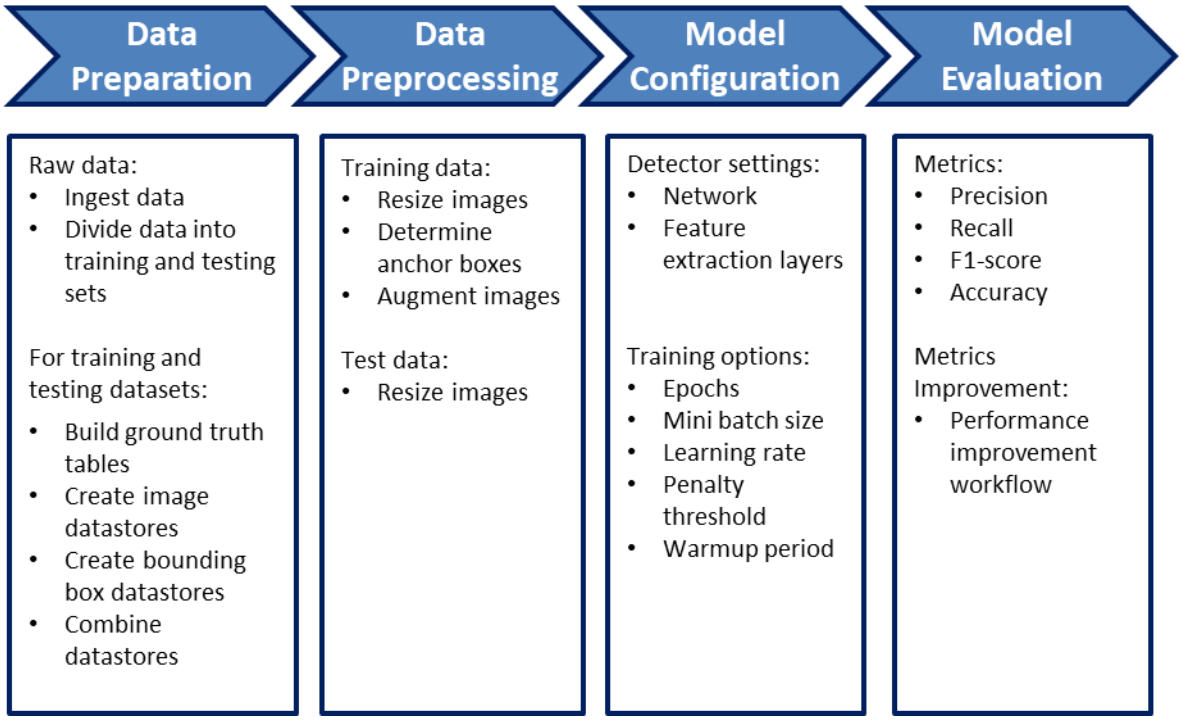

Figure 3.1: Model Training Methodology

3.1 Data Preparation

For supervised learning, a subject matter expert will take a set of data and using a labeling application, draw bounding boxes around objects of interest and assign them a class label. This set of data is known as a ground truth dataset. The raw ground truth dataset utilized in this study was provided to Old Dominion University by an industry partner in the form of images with corresponding xml files containing the bounding box and class label annotations.

3.1.1 Data Ingestion and Division

The xml files were ingested into the MATLAB® workspace and stored in a structure containing the name of the corresponding image files, the bounding box annotations, and defect class labels. A count of all object labels contained within the xml files was performed to determine the unique defect class labels used within the data along with their corresponding defect class count. The total available data was divided using a 70% and 30% split for training/testing purposes respectively by defect class count. The division process was not based on image file count as some images contain more defects than others and some defect classes are rare within the dataset. A summary of the available data and its division into training and test datasets is provided in Table 3.1. Although the data split was performed based on defect class count, the file division percentage was very close to 70% and 30% as captured in Table 3.2.

Defect Class	Training Data (70%)	Test Data (30%)	Total Data
Burr	714	306	1020
Cavity	463	199	662
Crackle	428	184	612
Curl Dregs	18	8	26
Dregs	1827	784	2611
Hole	626	269	895
Incomplete	147	64	211
Long Burr	28	12	40
Low Dregs	52	23	75
Shrinkage	884	380	1264
Side	4	2	6
Stripe Incomplete	7	3	10
Total	5,198	2,234	7,432

Table 3.1: Training and Test Data Summary

Data	Files	Percent
Training data	4019	69.74
Test data	1744	30.26
Total	5763	100

Table 3.2: Training and Test File Summary

3.1.2 Ground Truth Tables

To make the ingested data usable, a ground truth object is built for both the training and testing datasets. A ground truth object is a way of representing the ingested data in a usable format for model training and testing purposes. A sample of the ground truth bounding box

annotations is provided in Figure 3.2. The row numbers correspond to a unique image file available in a separate reference table within the ground truth object, and the columns correspond to any defect classes annotated within the image. The data contained within populated cells are coordinates of existing bounding boxes in the format of x, y, width, and height, where x and y are the coordinates of the top left corner of the bounding box. Each cell may have multiple entries if multiple defects of the same class are present. A blank cell represents a lack of a defect of the given class (column) for its respective image file (row).

gTruth_model.LabelData

	1 burr	2 cavity	3 crackle	4 curl_dregs	5 dregs	6 hole	7 incomplete	8 long_burr	9 low_dregs	10 shrinkage	11 side	12 stripe_incomplete
1044	[]	[]	[]	[]	[]	[902,128,31,...	[]	[]	[]	[]	[]	[]
1045	[]	[]	[347,568,31...	[]	[]	[]	[]	[]	[]	[]	[]	[]
1046	[]	[393,608,64,...	[]	[]	[]	[]	[]	[]	[]	[]	[]	[]
1047	[]	[336,710,61,...	[]	[]	[]	[432,331,58,...	[]	[]	[]	[]	[]	[]
1048	[]	[]	[384,522,24...	[]	[]	[]	[]	[]	[]	[]	[]	[]
1049	[]	[699,585,10...	[418,510,21...	[]	[]	[]	[]	[]	[]	[]	[]	[]
1050	[]	[]	[390,503,27...	[]	[]	[]	[]	[]	[]	[]	[]	[]
1051	[]	[]	[]	[]	[]	[280,232,49,...	[]	[]	[]	[]	[]	[]
1052	[]	[392,500,97,...	[]	[]	[]	[]	[]	[]	[]	[]	[]	[]
1053	[]	[]	[]	[]	[]	[468,274,46,...	[]	[]	[]	[]	[]	[]
1054	[]	[]	[]	[]	[]	[227,551,48,...	[]	[]	[]	[]	[]	[]
1055	[]	[51,84,618,2...	[]	[]	[]	[]	[]	[]	[]	[]	[]	[]
1056	[]	[]	[691,164,30...	[]	[]	[]	[]	[]	[]	[]	[]	[]
1057	[]	[]	[335,290,74,...	[449,439,95,80]	[]	[]	[]	[]	[]	[]	[]	[]
1058	[]	[]	[197,75,598,...	[]	[]	[]	[]	[]	[]	[]	[]	[]
1059	[]	[]	[]	[]	[]	[426,336,63,...	[]	[]	[]	[]	[]	[]
1060	[]	[]	[218,651,23...	[]	[]	[]	[]	[]	[]	[]	[]	[]
1061	[]	[]	[]	[]	[]	[658,82,31,28]	[]	[]	[]	[]	[]	[]
1062	[]	[]	[]	[]	[]	[471,72,35,25]	[]	[]	[]	[]	[]	[]
1063	[]	[]	[557,727,16...	[]	[]	[]	[]	[]	[]	[]	[]	[]
1064	[]	[226,372,48,...	[]	[]	[]	[]	[]	[]	[]	[]	[]	[]
1065	[]	[493,329,77,...	[]	[]	[]	[]	[]	[]	[]	[]	[]	[]
1066	[]	[]	[]	[]	[]	[396,222,29,...	[]	[]	[]	[]	[]	[]
1067	[]	[]	[]	[]	[]	[165,584,29,...	[]	[]	[]	[]	[]	[]
1068	[]	[]	[]	[]	[]	[796,604,26,...	[]	[]	[]	[]	[]	[]
1069	[]	[]	[]	[]	[]	[]	[]	[]	[]	[534,263,172,...	[]	[]
1070	[]	[]	[]	[]	[]	[]	[]	[]	[]	[451,114,123,...	[]	[]
1071	[]	[]	[]	[]	[]	[]	[]	[]	[]	[474,179,70,1...	[]	[]

Figure 3.2: Sample Ground Truth Data

The images associated with each dataset are moved into separate directories for later use during training and testing. From the ground truth objects, a bounding box datastore and an image datastore are built for both the training and testing datasets.

3.2 Data Preprocessing – Augmentation and Resize

Prior to training the YOLOv3 model, a data augmentation transformation is applied to the training dataset. This is especially helpful when training small datasets as it generates synthesized images for the model to train on, thus expanding the pool of training images without additional data collection (Shorten and Khoshgoftar, 2019). This practice also prevents the model from overfitting and improves precision. Table 3.3 contains a list of the image properties and their ranges that the data augmentation transformation applies randomly to each iteration. Properties without units are scalars. A sample output of this transformation is provided in Figure 3.3.

Property	Lower Limit	Upper Limit
Contrast	0.8	1.2
Saturation	0.9	1.1
Brightness	0.8	1.2
x-axis reflection (binary)	0	1
Rotation (degrees)	-15	15
	75	105
	165	195
	255	285

Table 3.3: Data Augmentation Properties

Any data augmentations, specifically rotations, resulting in a bounding box transformation with an area overlap of less than 0.6 were discarded. Post-augmentation, the data is then resized to the target preprocessing resolution of 256x256 pixels prior to training.

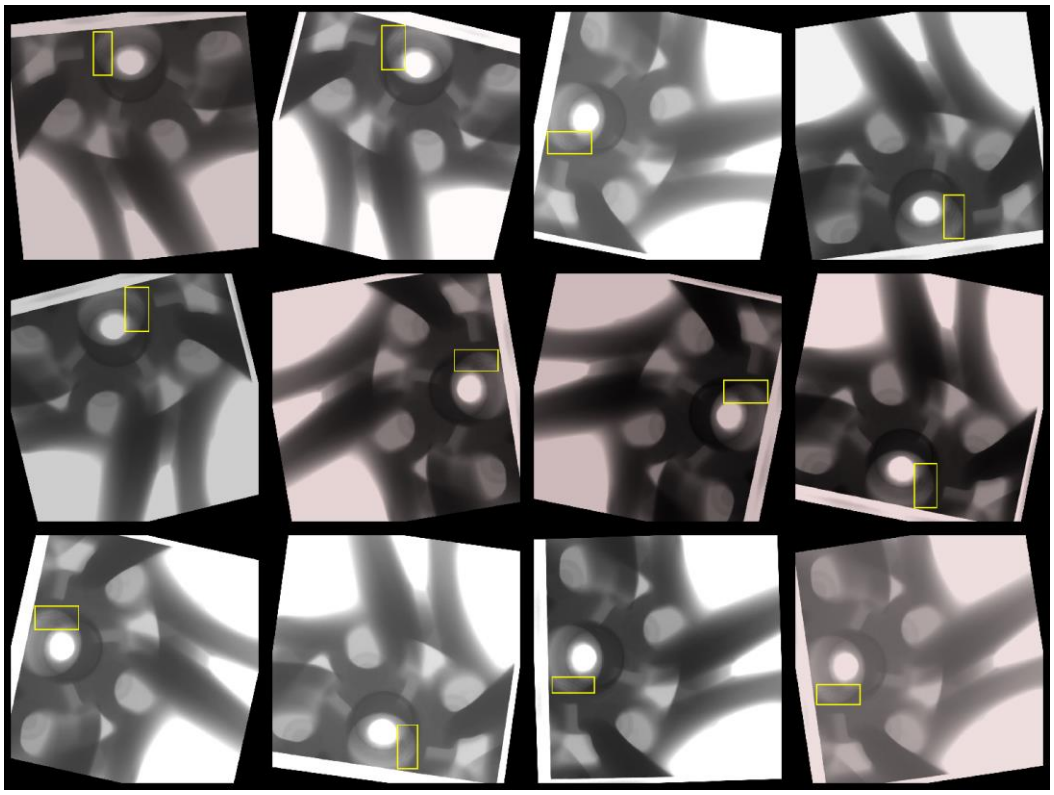

Figure 3.3: Sample Augmented Data

3.2.1 Determine Anchor Boxes

The number of anchor boxes to be used in the object detection model, and their aspect ratios, is determined by creating a training dataset for anchor box estimation. This dataset consists of a resize transformation without any augmentation to the desired training resolution. K-means clustering is used to capture the aspect ratios within the training dataset most likely to produce quality detections (Redmon and Farhadi, 2017). The mean intersection over union (IoU) achieved by the determined anchor box aspect ratios and the number of anchor boxes utilized can be estimated. A sample output of this function measured against the training dataset resized to a resolution of 256x256 pixels is shown in Figure 3.4.

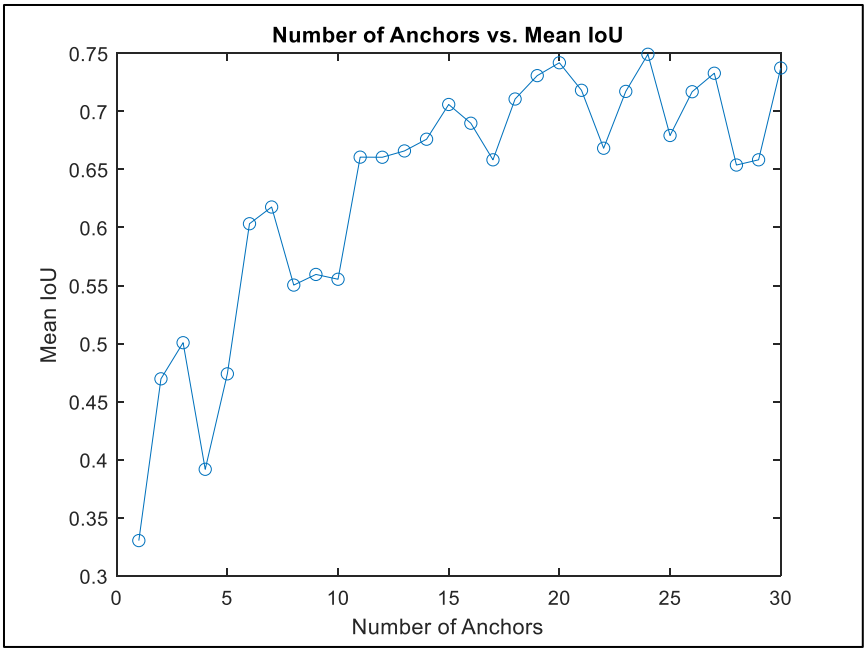

Figure 3.4: Number of Anchors vs. Mean IoU

The number of anchors used across all training efforts was set to a fixed value of six. Selecting a higher number of anchor boxes has diminishing returns as it increases computational costs and leads to overfitting. Figure 3.5 depicts the definition of intersection over union where one box represents the detected bounding box and the other represents the bounding box data from the ground truth dataset.

$$\text{Intersection over Union} = \frac{\text{Area of Overlap}}{\text{Area of Union}}$$

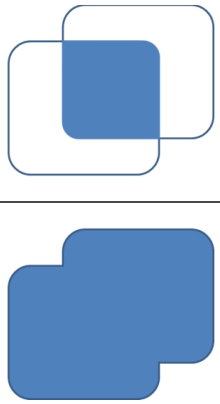

Figure 3.5: Intersection over Union

3.3 Model Configuration

Prior to model training, a feature extraction network is selected, and several training options are configured. This study focuses on evaluating the performance of Darknet-53 as the feature extraction network.

3.3.1 Training Options

There are many options available for configuration during training. The following five options were changed from their default values:

- Mini batch size – 30. The training dataset consists of 4,019 training images. A mini batch size of 30 allows the last partial mini batch to consist of 29 images.
- Learning rate – 0.001. This was experimentally determined to be optimal. A learning rate higher than 0.001 would result in the model loss diverging from 0. Conversely, a learning rate lower than 0.001 produced models that asymptotically approached a higher model loss value than the loss produced by the models with a learning rate of 0.001.
- Penalty threshold – 0.5. Detections that overlap less than 50% with the ground truth bounding box training data are penalized.
- Warmup period – 1000 iterations. A warmup period is applied to prevent the model from overfitting early in the training cycle. This period helps with stabilizing gradients when using higher learning rates.
- Epochs – variable. An epoch is defined as the number of passes of the entire training dataset. With a mini batch size of 30, an epoch requires 134 iterations to cycle through the training dataset once. The model is allowed to continue training as long as the average

loss continues to decrease. The highest performing model completed training in 55 epochs.

3.4 Model Evaluation

To streamline the discussion of metrics in this section and to help avoid repetition, the more common components across formulas and their definitions will be presented upfront. When speaking of failed or successful predictions by the trained model, these are all within the context of the data contained within the ground truth test dataset.

- TP – True Positives – correct detections of existing defects
- FP – False Positives – incorrect detections of non-existent defects
- TN – True Negatives – correct failures to detect non-existent defects
- FN – False Negatives – incorrect failures to detect existing defects

3.4.1 Metrics

The following metrics were used to evaluate the performance of the defect detection models. The definitions for precision, recall, F1-score, and accuracy were adopted from the National Institute of Standards and Technology (2014).

Precision – fraction of true positives predictions relative to the sum of true positives and false positives. Precision is the fraction of positive predictions that are correct.

$$Precision = \frac{TP}{TP + FP} \quad (1)$$

Recall – fraction of true positives relative to the sum of true positives and false negatives. Recall is the fraction of detected defects relative to the number of defects that should have been detected.

$$Recall = \frac{TP}{TP + FN} \tag{2}$$

F1-score – a measure of the of the model's accuracy. The values of the F1-score are limited between 0 and 1. The score will be 0 if precision and or recall are 0, and the score will be 1 when precision and recall are both 1. It is defined as the harmonic mean of the model's precision and recall (Padilla et al., 2021).

$$F1 = 2 * \frac{precision * recall}{precision + recall} \tag{3}$$

Accuracy – the sum of true positives and true negatives relative to the total number of detections. Accuracy measures the fraction of correct detections relative to the total number of detections.

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \tag{4}$$

3.4.2 Metrics Improvement

The improvement of model performance metrics is performed iteratively by following the workflow based on the IBM® (2017) model analysis of precision and recall in Figure 3.6.

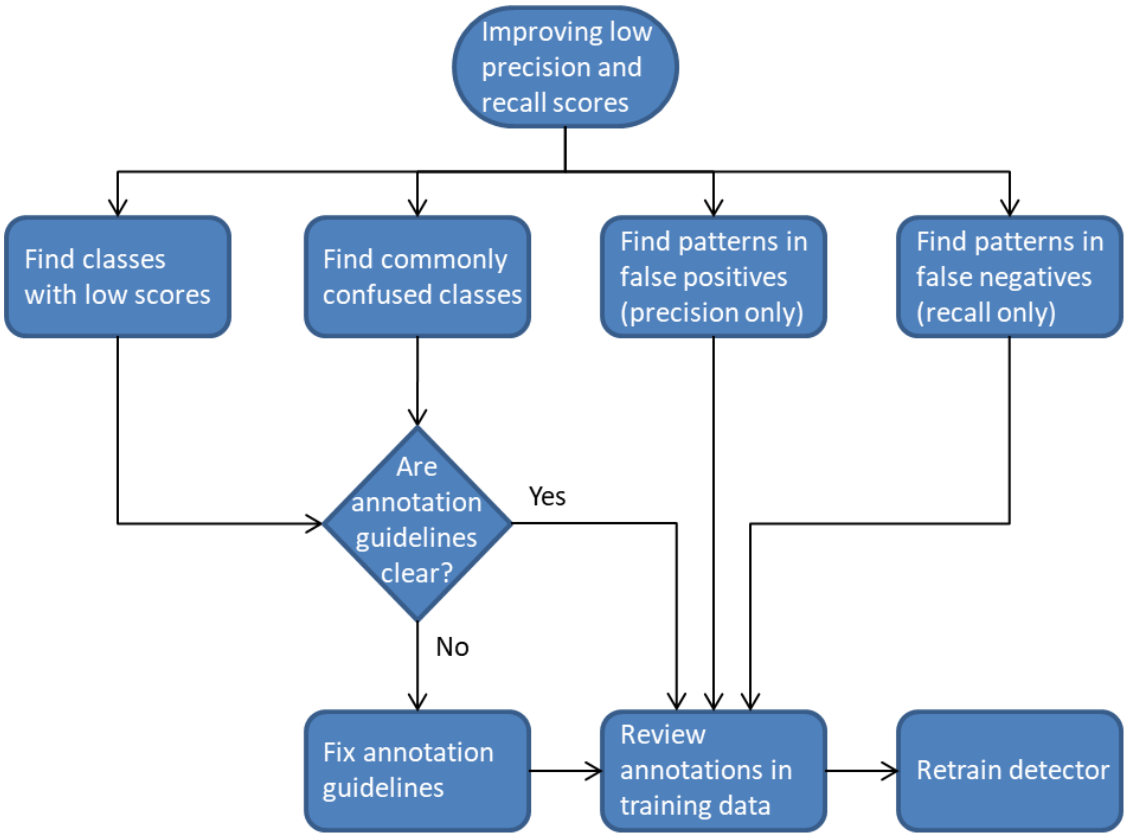

Figure 3.6: Precision and Recall Improvement Workflow

In the workflow, annotations refer to ground truth class labels provided by the tire and rim manufacturer. Old Dominion University was not involved in process of defining labeling guidelines; however, in any classes with low scores and/or confused classes it can be inferred that the annotation guidelines were suboptimal. Making these changes with the manufacturer was not within the scope of this study. These issues will be remedied through class relabeling and class combination for optimal detector performance. The specific practices implemented as related to class relabeling and combination are discussed in detail in chapter 4.

3.4.3 Confused Defect Classes

To improve performance metrics using the workflow in Figure 3.6, confused classes need to be identified. To achieve this, a confusion matrix is generated for each detection model. A confusion matrix characterizes the performance of the classification portion of the object detection algorithm at a given IoU. To gain insight across all possible classifications made by the detection model, the IoU threshold will be set to a value of greater than zero.

3.5 Hardware and Software Environment

Table 3.4 provides a summary of the hardware environment. The software used was MATLAB® R2021b.

System Configuration	
GPU	NVIDIA RTX™ 3090 (24GB VRAM)
CPU	Intel® Core™ i7-87000K
RAM	32 GB
Operating System	Windows 10 64-bit

Table 3.4: Hardware Environment

CHAPTER 4

RESULTS

Following the training completion of the defect detection model, the defect detector was evaluated against the test dataset that was set aside during the data preparation phase. Performance analysis of the original defect detector is provided in this chapter along with the analysis of subsequently trained models based on the performance improvement workflow discussed in section 3.4.2.

4.1 Original Defect Detector Performance

To assess model performance, the precision, recall, and F1-score for each defect class was calculated along with the overall model accuracy at different IoU thresholds ranging from 0.5 to 0 in increments of 0.1. The results for precision, recall, F1-score, and accuracy are available in Table 4.1, Table 4.2, Table 4.3 and Table 4.4 respectively. As the IoU threshold is lowered, the performance metrics increase as the detector is interpreting detections with less area overlap requirements as positive detections.

It is worth noting that for the application in this study, maximizing the detection of defects is of higher priority than knowing precisely where a defect is located. The emphasis on the performance of the defect detector is placed on its ability to correctly identify as many defective products as possible and thus remove them from production. Assessment of the detector can be focused around an IoU threshold of greater than zero. The performance of the detector at an IoU greater than zero is captured in the form of a confusion matrix in Figure 4.1 and is also tabulated in previously mentioned tables. The confusion matrix allows for the identification of confused classes during classification.

To aid with interpretation of the confusion matrix, the far-right column corresponds to the precision per defect class, the bottom row corresponds to recall per class, and the bottom right cell represents the model's accuracy.

IoU	Precision (%)												
	Burr	Cavity	Crackle	Curl Dregs	Dregs	Hole	Incomplete	Long Burr	Low Dregs	Shrinkage	Side	Stripe	Model Precision
0.5	33.2	27.1	39.3	60.0	71.1	24.2	59.7	100	15.4	32.5	50.0	0	46.3
0.4	39.2	30.8	49.0	80.0	81.0	33.6	69.4	100	35.7	40.5	50.0	0	54.7
0.3	45.3	36.7	56.3	90.0	85.0	47.8	79.0	100	42.9	46.8	50.0	0	61.3
0.2	52.1	39.4	64.3	90.0	88.0	55.7	84.1	100	57.1	52.1	50.0	0	66.3
0.1	58.1	43.0	69.1	100	89.5	61.3	87.5	100	66.7	54.1	50.0	0	69.6
> 0	62.1	44.2	71.3	100	89.7	63.1	89.1	100	75.0	54.3	50.0	0	70.9

Table 4.1: Original Model – Precision

IoU	Recall (%)												
	Burr	Cavity	Crackle	Curl Dregs	Dregs	Hole	Incomplete	Long Burr	Low Dregs	Shrinkage	Side	Stripe	Model Recall
0.5	28.1	42.0	43.4	54.5	73.1	25.9	60.7	57.1	8.3	42.0	100	0	49.5
0.4	33.0	47.3	54.5	72.7	81.8	36.0	68.3	57.1	18.5	51.3	100	0	57.8
0.3	37.7	55.3	62.0	81.8	85.4	51.3	76.6	57.1	22.2	58.7	100	0	64.4
0.2	42.2	58.9	69.2	81.8	89.8	59.8	81.5	57.1	29.6	64.6	100	0	69.8
0.1	45.8	63.9	75.2	90.9	91.7	64.1	86.2	57.1	44.4	66.7	100	0	73.1
> 0	47.0	66.0	80.5	90.9	92.6	66.2	87.7	57.1	62.1	67.2	100	0	74.7

Table 4.2: Original Model – Recall

IoU	F1-score (%)												
	Burr	Cavity	Crackle	Curl Dregs	Dregs	Hole	Incomplete	Long Burr	Low Dregs	Shrinkage	Side	Stripe	Model F1-score
0.5	30.4	32.9	41.3	57.1	72.1	25.0	60.2	72.7	10.8	36.6	66.7	0	47.8
0.4	35.8	37.3	51.6	76.2	81.4	34.8	68.9	72.7	24.4	45.3	66.7	0	56.2
0.3	41.2	44.1	59.0	85.7	85.2	49.5	77.8	72.7	29.3	52.1	66.7	0	62.8
0.2	46.6	47.2	66.7	85.7	88.9	57.7	82.8	72.7	39.0	57.7	66.7	0	68.0
0.1	51.2	51.4	72.0	95.2	90.6	62.7	86.9	72.7	53.3	59.7	66.7	0	71.3
> 0	53.5	52.9	75.6	95.2	91.1	64.6	88.4	72.7	67.9	60.1	66.7	0	72.8

Table 4.3: Original Model – F1-score

IoU	Overall Performance (%)			
	Precision	Recall	F1	Accuracy
0.5	46.3	49.5	47.8	43.3
0.4	54.7	57.8	56.2	51.6
0.3	61.3	64.4	62.8	58.7
0.2	66.3	69.8	68.0	64.8
0.1	69.6	73.1	71.3	68.9
> 0	70.9	74.7	72.8	70.7

Table 4.4: Original Model – Overall Performance

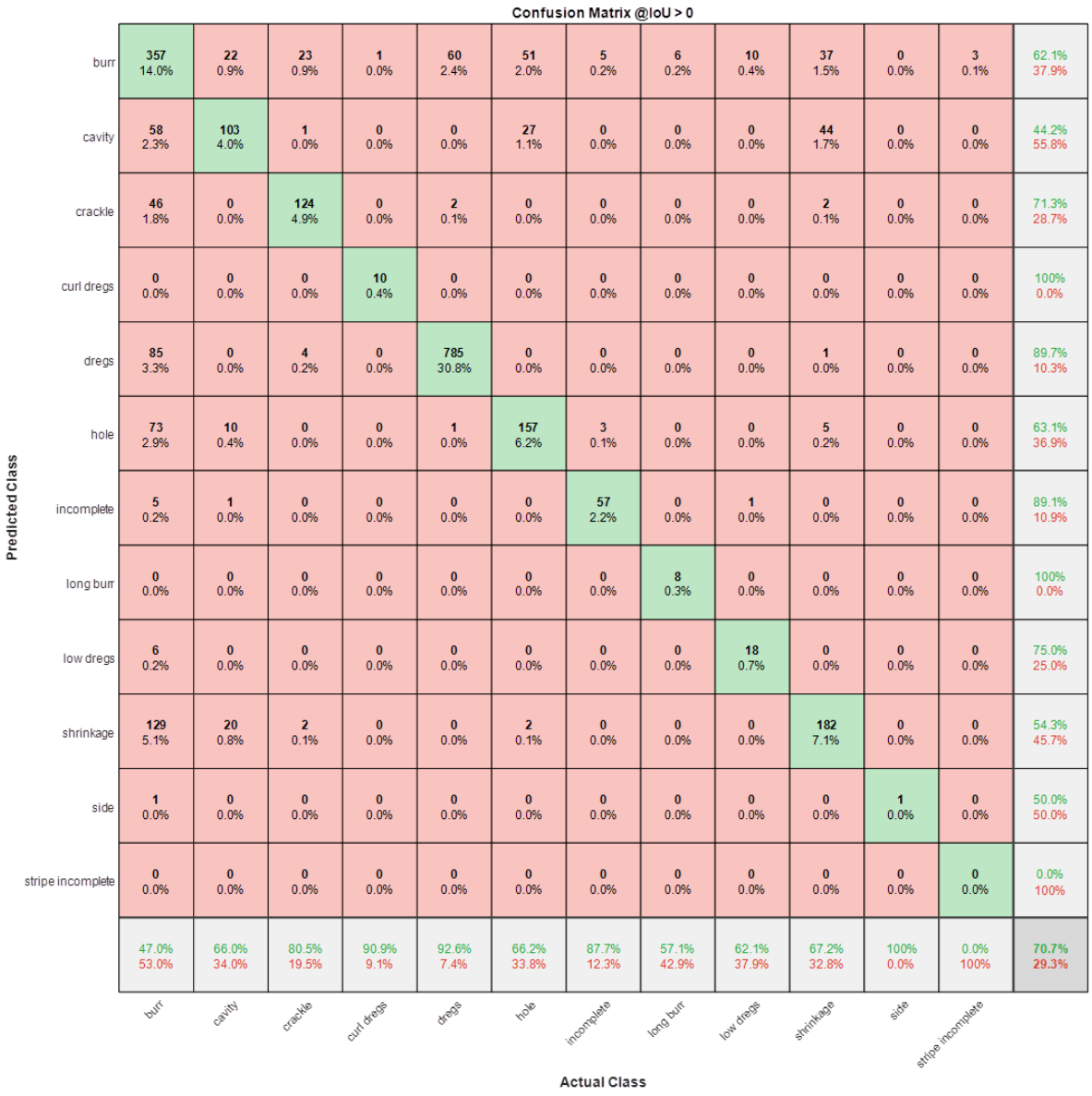

Figure 4.1: Confusion Matrix – Original Model

Analysis of the confusion matrix shows that the detection results for the defect classes burr, cavity, crackle, hole, low dregs, and shrinkage contain a large amount of incorrect predicted classes versus the actual classes contained in the ground truth test dataset. A summary of this data is provided in Table 4.5.

The goal of the work in this thesis is to produce a model capable of detecting as many defective products as possible and removing them from production, not necessarily identifying the exact defect class present. With that in mind, training images pertaining to the defect classes in Table 4.5 were examined for commonalities and potential for class combination. Cavity, crackle, hole, low dregs, and shrinkage are all visually similar. They generally appear as solitary, or clusters of, small circular geometries that are lighter than the background object in the radiograph. Burr, on the other hand, shares the same physical characteristics but is visually darker than the background object.

Predicted	Actual					
	burr	cavity	crackle	hole	low dregs	shrinkage
burr	357	22	23	51	10	37
cavity	58	103	-	27	-	44
crackle	46	-	124	-	-	-
hole	73	-	-	157	-	-
low dregs	6	-	-	-	23	-
shrinkage	129	20	-	-	-	182

Table 4.5: Confused Defect Classes

4.2 Cavity and Burr Detector Performance

Given the high number of confused classes, the precision and recall improvement workflow discussed in section 3.4.2 was followed to improve model performance. To fix the defect label annotations, a new set of training and test datasets based off the original ground truth tables was constructed where any defects labeled as crackle, hole, low dregs, and shrinkage were relabeled as cavity. The combination of similar classes is a practice known as data relabeling where classes appearing to belong to a hierarchical super class can be combined into one class

(Lee et al., 2018). A new model was subsequently trained with the newly combined data while maintaining the original training configurations discussed in section 3.3. This model, referred to as the cavity and burr model, continued to show that burr and the new cavity super class remained confused. The confusion matrix for the cavity and burr model is available in Figure 4.2.

Even though new cavity super class and the burr defect class continued to be confused, the performance of the cavity and burr model exceeded the performance of the original model across all metrics. Similar to how the performance of the original model was characterized across several IoU levels, the same has been done for the cavity and burr model. The results for precision, recall, F1-score, and accuracy are available in Table 4.6, Table 4.7, Table 4.8 and Table 4.9 respectively.

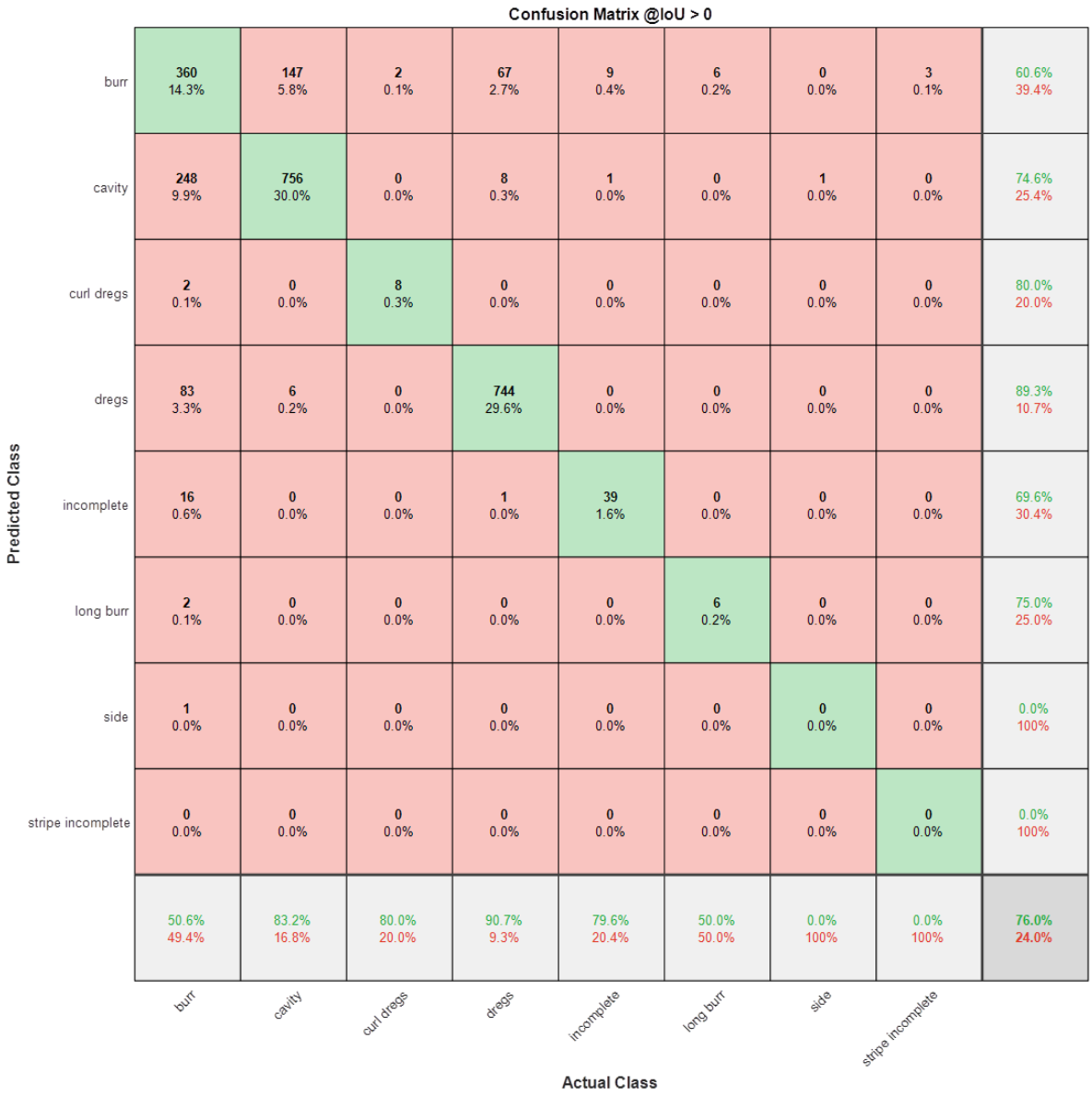

Figure 4.2: Confusion Matrix – Cavity and Burr Model

IoU	Precision (%)								
	Burr	Cavity	Curl Dregs	Dregs	Incomplete	Long Burr	Side	Stripe	Model Precision
0.5	32.5	38.7	40.0	68.1	38.2	42.9	0	0	48.1
0.4	38.8	50.1	60.0	80.1	57.4	75.0	0	0	59.4
0.3	45.2	61.2	70.0	86.3	61.1	75.0	0	0	67.8
0.2	50.7	68.5	80.0	87.9	65.5	75.0	0	0	72.7
0.1	55.5	72.3	80.0	89.2	67.9	75.0	0	0	75.7
> 0	60.6	74.6	80.0	89.3	69.6	75.0	0	0	77.6

Table 4.6: Cavity and Burr Model – Precision

IoU	Recall (%)								
	Burr	Cavity	Curl Dregs	Dregs	Incomplete	Long Burr	Side	Stripe	Model Recall
0.5	29.9	43.4	44.4	67.4	43.8	27.3	0	0	49.8
0.4	35.6	55.7	60.0	78.8	66.0	50.0	0	0	61.2
0.3	41.5	67.2	70.0	84.2	70.2	50.0	0	0	69.5
0.2	46.1	74.5	80.0	87.0	73.5	50.0	0	0	74.7
0.1	48.8	78.7	80.0	90.2	77.6	50.0	0	0	78.3
> 0	50.6	83.2	80.0	90.7	79.6	50.0	0	0	80.9

Table 4.7: Cavity and Burr Model – Recall

IoU	F1-score (%)								
	Burr	Cavity	Curl Dregs	Dregs	Incomplete	Long Burr	Side	Stripe	Model F1-score
0.5	31.2	40.9	42.1	67.8	40.8	33.4	0	0	48.9
0.4	37.1	52.8	60.0	79.4	61.4	60.0	0	0	60.3
0.3	43.3	64.1	70.0	85.2	65.3	60.0	0	0	68.6
0.2	48.3	71.4	80.0	87.5	69.3	60.0	0	0	73.7
0.1	51.9	75.4	80.0	89.7	72.4	60.0	0	0	77.0
> 0	55.2	78.7	80.0	90.0	74.3	60.0	0	0	79.2

Table 4.8: Cavity and Burr Model – F1-score

IoU	Overall Performance (%)			
	Precision	Recall	F1	Accuracy
0.5	48.1	49.8	48.9	43.8
0.4	59.4	61.2	60.3	54.8
0.3	67.8	69.5	68.6	63.7
0.2	72.7	74.7	73.7	69.5
0.1	75.7	78.3	77.0	73.4
> 0	77.6	80.9	79.2	76.0

Table 4.9: Cavity and Burr Model – Overall Performance

4.3 Cavity Detector Performance

Although defects belonging to the burr class have visual distinctions from the new cavity super class, they share enough characteristics that the model has difficulty distinguishing them. Like the process where four of the defect classes were absorbed into the cavity super class, the same procedure was applied to all burr defects. All defects labeled as burr were relabeled as cavity, new training and test datasets were constructed, and a new model was trained. The new model, referred to as the cavity model, performed significantly better than the two previous models. The high class confusion has also been resolved. This can be seen in the confusion matrix of the cavity model in Figure 4.3.

The cavity model's precision, recall, F1-score, and accuracy are captured in Table 4.10, Table 4.11, Table 4.12, and Table 4.13 respectively.

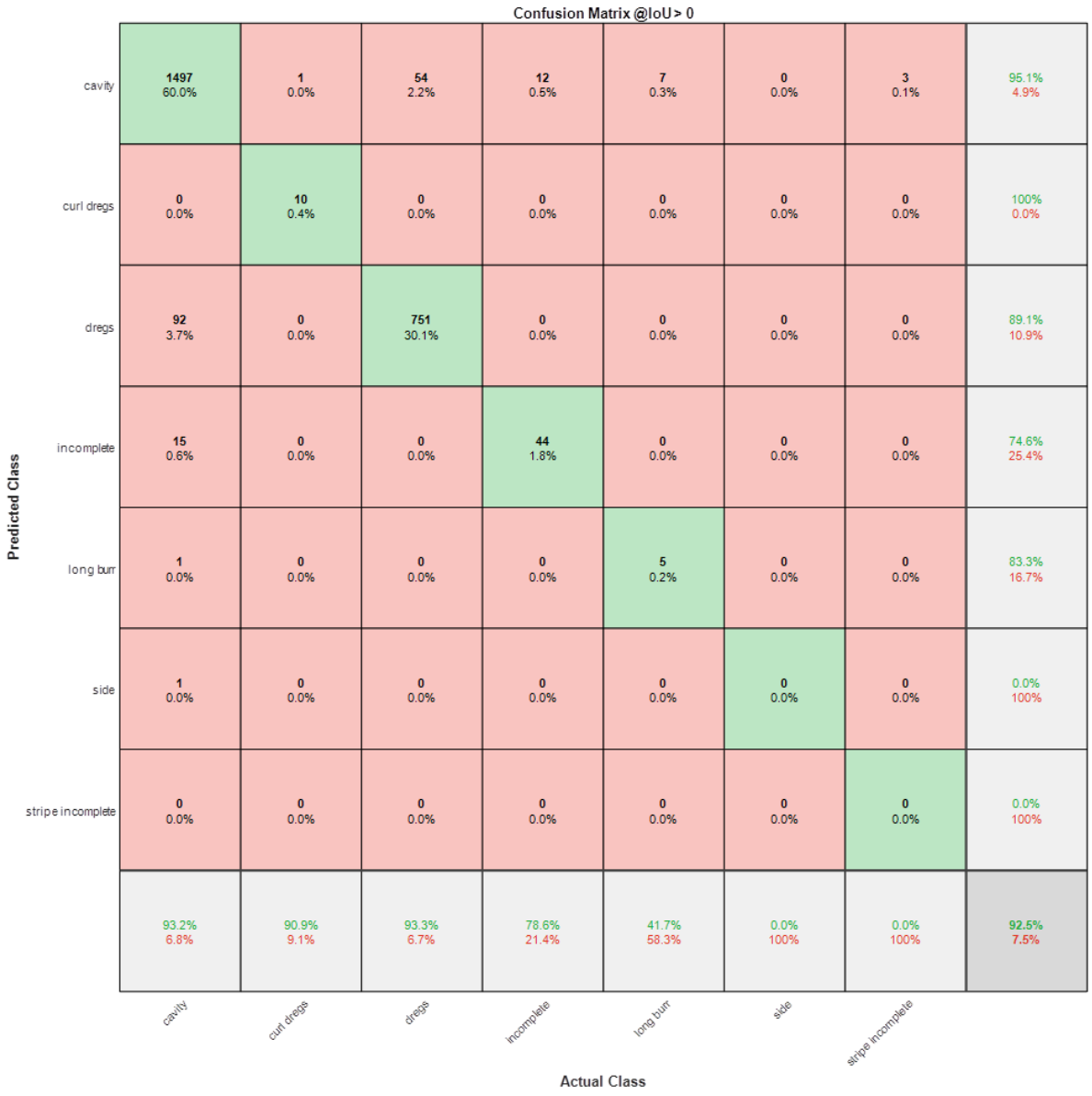

Figure 4.3: Confusion Matrix – Cavity Model

IoU	Precision (%)							
	Cavity	Curl Dregs	Dregs	Incomplete	Long Burr	Side	Stripe	Model Precision
0.5	86.5	90.0	68.1	40.4	60.0	0	0	78.4
0.4	89.6	90.0	78.4	57.1	83.3	0	0	84.5
0.3	90.7	90.0	83.8	69.9	83.3	0	0	87.4
0.2	92.8	100	87.0	71.9	83.3	0	0	89.9
0.1	94.7	100	88.9	74.1	83.3	0	0	91.8
> 0	95.1	100	89.1	74.6	83.3	0	0	92.2

Table 4.10: Cavity Model – Precision

IoU	Recall (%)							
	Cavity	Curl Dregs	Dregs	Incomplete	Long Burr	Side	Stripe	Model Recall
0.5	86.4	81.8	68.5	45.1	27.3	0	0	78.4
0.4	89.7	81.8	78.6	60.4	41.7	0	0	84.5
0.3	91.6	81.8	82.6	70.9	41.7	0	0	87.3
0.2	92.8	90.9	87.8	73.2	41.7	0	0	90.0
0.1	93.2	90.9	92.4	76.8	41.7	0	0	92.0
> 0	93.2	90.9	93.3	78.6	41.7	0	0	92.3

Table 4.11: Cavity Model – Recall

IoU	F1-score (%)							
	Cavity	Curl Dregs	Dregs	Incomplete	Long Burr	Side	Stripe	Model F1-score
0.5	86.5	85.7	68.3	42.6	37.5	0	0	78.4
0.4	89.7	85.7	78.5	58.7	55.6	0	0	84.5
0.3	91.2	85.7	83.2	70.4	55.6	0	0	87.4
0.2	92.8	95.2	87.4	72.5	55.6	0	0	90.0
0.1	93.9	95.2	90.6	75.4	55.6	0	0	91.9
> 0	94.1	95.2	91.2	76.6	55.6	0	0	92.2

Table 4.12: Cavity Model – F1-score

IoU	Overall Performance (%)			
	Precision	Recall	F1	Accuracy
0.5	78.4	78.4	78.4	80.7
0.4	84.5	84.5	84.5	85.7
0.3	87.4	87.3	87.4	88.1
0.2	89.9	90.0	90.0	90.5
0.1	91.8	92.0	91.9	92.3
> 0	92.2	92.3	92.2	92.5

Table 4.13: Cavity Model – Overall Performance

4.4 Overall Detector Performance

To make comparisons across models easier, the overall model performance values for precision, recall, F1-score, and accuracy have been plotted against their respective IoU values in Figure 4.4, Figure 4.5, Figure 4.6, and Figure 4.7 respectively.

All detectors were capable of processing a test image resized to a resolution of 256x256 pixels in less than 90 milliseconds.

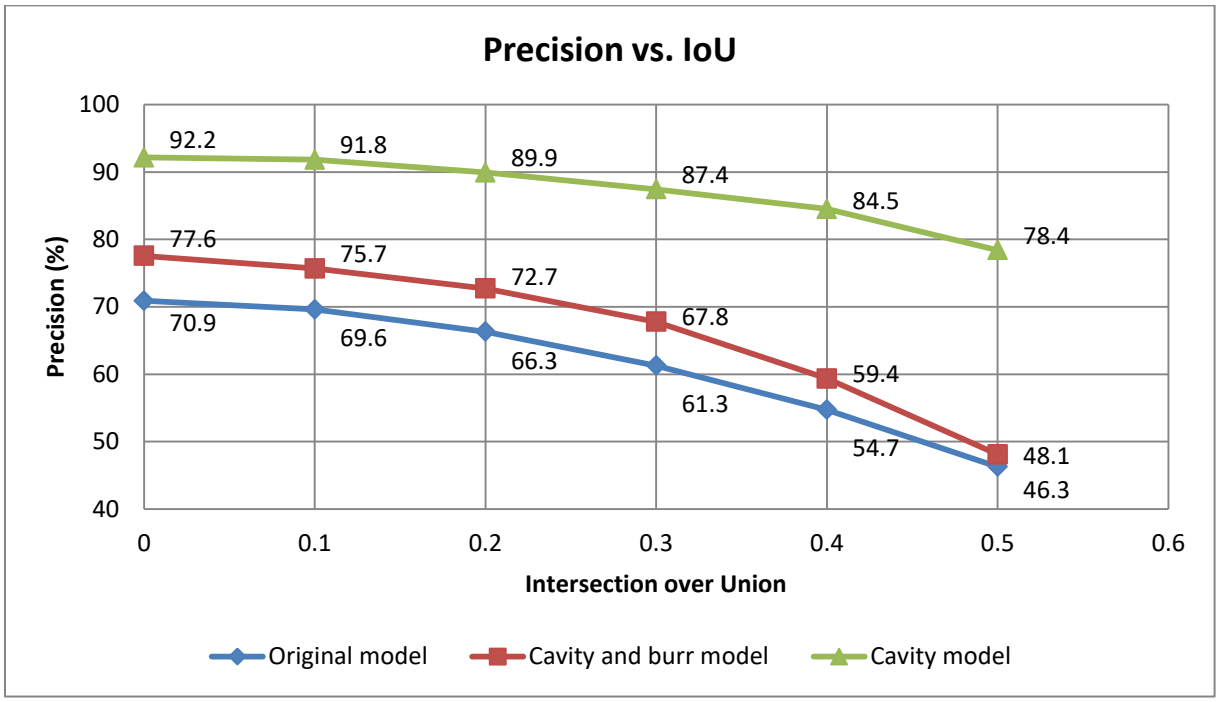

Figure 4.4: Overall Detector Performance – Precision

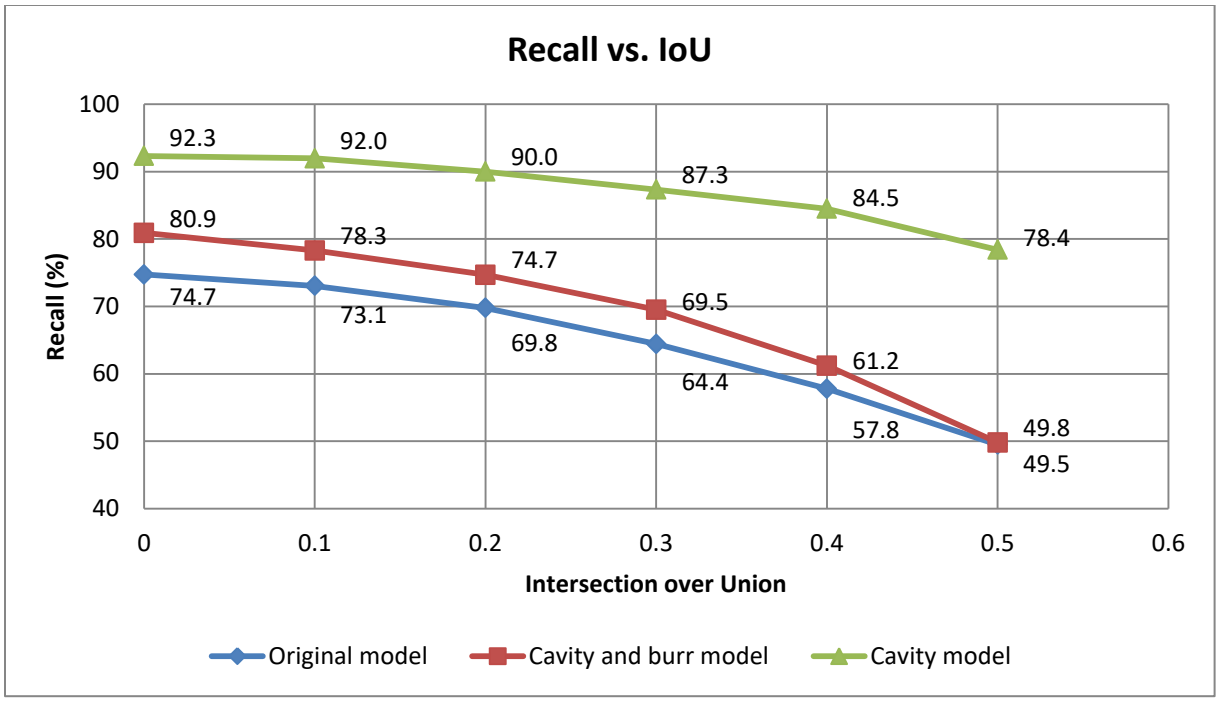

Figure 4.5: Overall Detector Performance – Recall

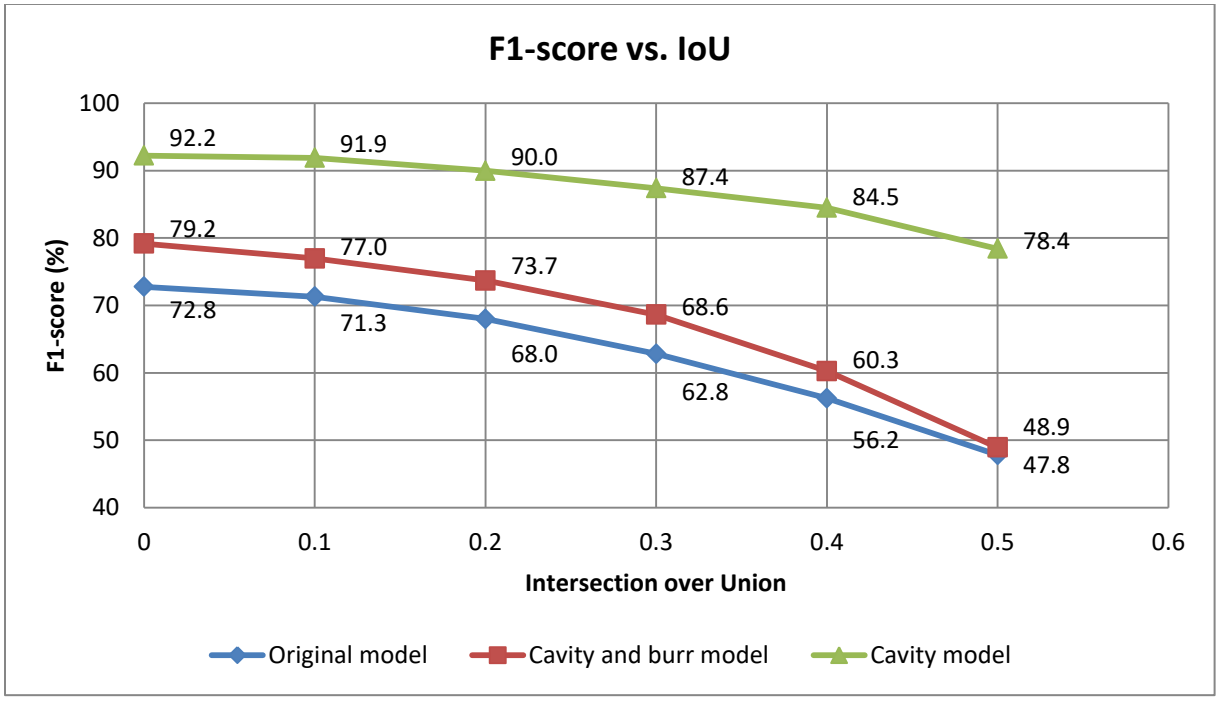

Figure 4.6: Overall Detector Performance – F1-score

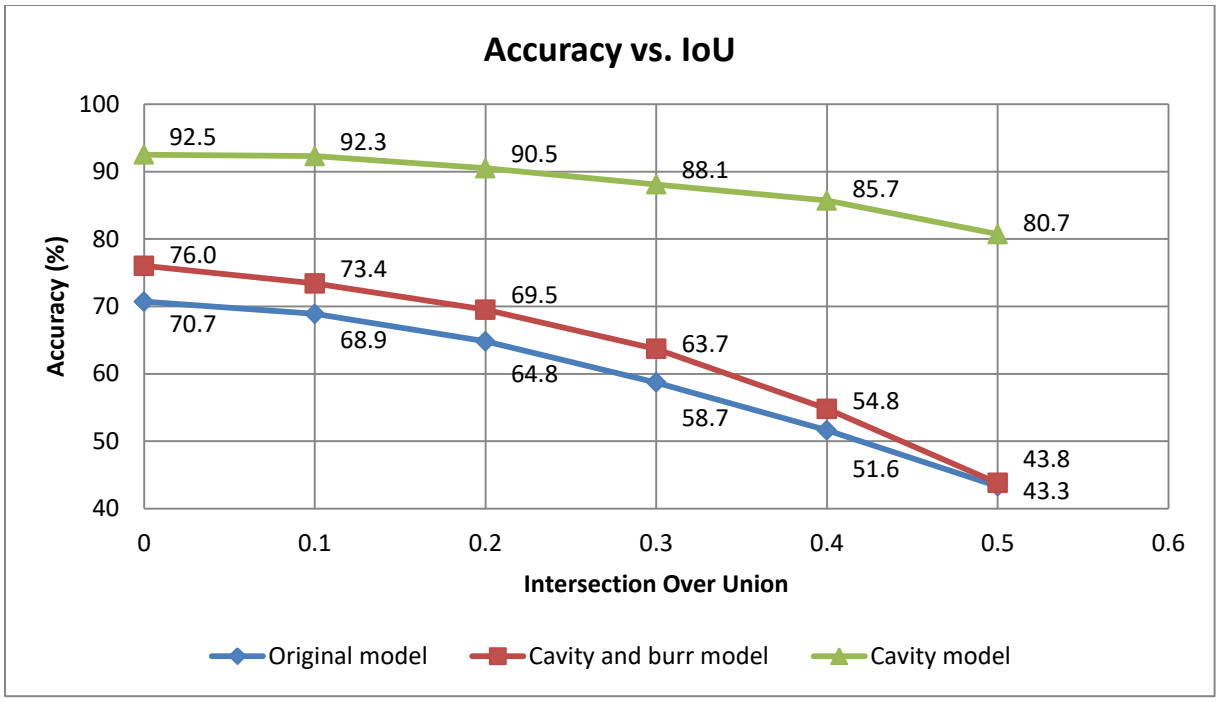

Figure 4.7: Overall Detector Performance – Accuracy

CHAPTER 5

CONCLUSIONS AND RECOMMENDATIONS

5.1 Conclusions

After applying data relabeling techniques to the original data, the performance of the object detector model improved significantly. In addition to selecting the appropriate object detector and detector network, it is important to establish and carefully follow data labeling guidelines as inconsistencies within the data annotations reduce model performance. Some of these inconsistencies and their detections by the model are included in the Appendix. The identification of defect classes to be tracked from a business perspective should be performed in a manner that is consistent with practices that are machine learning friendly. Defining defect classes that are similar to others (e.g., cavity, hole) can negatively impact model performance and lead to class confusion, as was the case in this study.

5.2 Recommendations

The following recommendations are being made with the goal of removing errors and challenges inherent within the ground truth dataset:

1. To help improve model performance metrics, establishing defect definition and annotation guidelines prior to labeling data with bounding boxes and defect class labels would help remove inconsistencies within the data.
2. Working with the manufacturer to approach the problem through the lens of a machine learning exercise where sufficiently distinct defect classes are categorized would help increase defect detection rate with a higher degree of confidence. For example, instead of

having multiple defect classes that are a sub-class of a much larger defect type (e.g., cavity, hole) a single comprehensive class should be assigned to similar defects.

Recommendations for future work after the previous recommendations have been completed include the following:

1. The models trained in this study utilized augmented images resized to a resolution of 256x256 pixels. Resolution of future models could be increased.
2. Training an object detector based on a two-stage detection method and comparing its performance against the YOLOv3 models generated in this study.

REFERENCES

Alzubaidi, L., Zhang, J., Humaidi, A., Al-Dujaili, A., Duan, Y., Al-Shamma, O., Santamaria, J., Fadhel, M. A., Al-Amidie, M., & Farhan, L. (2021). Review of deep learning: concepts, CNN, architectures, challenges, applications, future directions. *Journal of Big Data, 8*(53). https://doi.org/10.1186/s40537-021-00444-8

Aurich, J. C., Dornfeld, D., Arrazola, P. J., Franke, V., Leitz, & L., Min, S. (2009). Burrs – analysis, control and removal. *CIRP Annals, 58*(3), 519-542. https://doi.org/10.1016/j.cirp.2009.09.004

Choi, R., Coyner, A. S., Kalpathy-Cramer, J., Chiang, M. F., & Campbell, J. P. (2020). Introduction to machine learning, neural networks, and deep learning. *Translational Vision, Science & Technology, 9*(2). https://www.ncbi.nlm.nih.gov/pmc/articles/PMC7347027/

Deep AI. (2019, May 17). *Synapse,* https://deepai.org/machine-learning-glossary-and-terms/synapse

Deng, J., Xuan, X., Wang, W., Li, Z., Yao, H., & Wang, Z. (2020). A review of research on object detection based on deep learning. *Journal of Physics: Conference Series, 168*. https://doi.org/10.1088/1742-6596/1684/1/012028

Fadel, N., Al-Hameed, W., & Ahmed, M.K. (2020). Non destructive testing for detection abnormal object in the x-ray images. *Journal of Physics: Conference Series, 1660*. https://doi.org/10.1088/1742-6596/1660/1/012104

IBM (2017 August 14). *Analyzing machine-learning model performance.* https://cloud.ibm.com/docs/knowledge-studio?topic=knowledge-studio-evaluate-ml

IBM (2017 August 14). *Machine-learning model creation workflow.* https://cloud.ibm.com/docs/knowledge-studio?topic=knowledge-studio-ml_annotator

Ingle, V. (2017). Defects, root causes in casting process and their remedies: review. *International Journal of Engineering Research and Application, 7*(3), 47-54. https://doi.org/10.9790/9622-0703034754

Juriani, A. (2015). Casting defects in foundry and their remedial measures with industrial case studies. *IOSR Journal of Mechanical and Civil Engineering (IOSR-JMCE), 12*(6), 43-54. https://www.iosrjournals.org

LeCun, Y., Bengio, Y., & Hinton, G. (2015). Deep learning. *Nature*, 521, 436-444. https://doi.org/10.1038/nature14539

Lee, K. [Kibok], Lee., K. [Kimin], Min, K., Zhang, Y., Shin, J., & Lee, H. (2018). Hierarchical novelty detection for visual object recognition. *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition* (pp. 1034-1042). IEEE. https://doi.org/10.1109/CVPR.2018.00114

Liu, M., Chen, Y., He, L., Zhang, Y., & Xie, J. (2021). LF-YOLO: A lighter and faster YOLO for weld defect detection of x-ray image (arXiv:2110.15045). arXivLabs. https://arxiv.org/abs/2110.15045

Massachusetts Institute of Technology. (2021, April 21). *Machine learning explained.* https://mitsloan.mit.edu/ideas-made-to-matter/machine-learning-explained

McKinsey & Company. (2014). A future that works: Automation, employment, and productivity. https://www.mckinsey.com/~/media/mckinsey/featured%20insights/Digital%20Disruption/Harnessing%20automation%20for%20a%20future%20that%20works/MGI-A-future-that-works-Executive-summary.ashx

National Institute of Standards and Technology. (2014). *Performance metrics for evaluating object and human detection tracking systems* (NISTIR 7972). https://doi.org/10.6028/NIST.IR.7972

Nguyen, N-D., Do, T., Ngo, T. D., & Le, D-D. (2020). An evaluation of deep learning methods for small object detection. *Journal of Electrical and Computer Engineering*, 3189691. https://doi.org/10.1155/2020/3189691

Padilla, R., Passos, W. L., Dias, T. L., Netto, S. L., & da Silva, E. A. B. (2021). A comparative analysis of object detection metrics with a companion open-source toolkit. *Electronics, 10*(3). https://doi.org/10.3390/electronics10030279

Patil, R. T., Metri, V. S., Tambore, S. S. (2015). Causes of casting defects with remedies. *International Journal of Engineering Research & Technology (IJERT), 4*(11), 639-644. https://doi.org/10.17577/IJERTV4IS110511

Rai, P. K., Mohammad, A., & Jafri, H. Z. (2013). Causes & preventation of defects (burr) in sheet metal component. *International Journal of Engineering Research and Applications (IJERA), 3*(4), 511-515. https://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.414.7552&rep=rep1&type=pdf

Ralasic, I. (2021, October 11). *A better mAP for object detection.* Towards Data Science. https://towardsdatascience.com/a-better-map-for-object-detection-32662767d424

Redmon, J., Divvala, S., Girshick, R., & Farhadi, A. (2016). You only look once: Unified, real-time object detection. *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (pp. 779-788). IEEE. https://doi.org/10.1109/CVPR.2016.91

Redmon, J., & Farhadi, A. (2017). YOLO9000: Better, faster, stronger. *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (pp. 6517 – 6525). IEEE. https://doi.org/10.1109/CVPR.2017.690

Redmon, J., & Farhadi, A. (2018). YOLOv3: *An incremental improvement* (arXiv:1804.02767). arXivLabs. https://arxiv.org/abs/1804.02767

Ren, J., Ren, R., Green, M., & Huang, X. (2019, May 5-8). *Defect detection from x-ray images using a three-stage deep learning algorithm* [Paper presentation]. 2019 IEEE Canadian Conference of Electrical and Computer Engineering (CCECE), Edmonton, AB, Canada. https://doi.org/10.1109/CCECE.2019.8861944

Rocca, B. (2019, January 27). *Handling imbalanced datasets in machine learning.* Towards Data Science. https://towardsdatascience.com/handling-imbalanced-datasets-in-machine-learning-7a0e84220f28

Saeed, F., Ahmed, M. J., Gul, M. J., Hong, K. J., Paul, A., & Kavitha, M. S. (2021). A robust approach for industrial small-object detection using an improved faster regional convolutional neural network. *Scientific Reports, 11,* 23390 (2021). https://doi.org/10.1038/s41598-021-02805-y

Sarker, I. H. (2021). Machine Learning: Algorithms, real-world applications and research directions. *SN Computer Science, 160*(2). https://doi.org/10.1007/s42979-021-00592-x

Shorten, C., & Khoshgoftaar, T. M. (2019). A survey on image data augmentation for deep learning. *Journal of Big Data, 6*(60). https://doi.org/10.1186/s40537-019-0197-0

Solawetz, J. (2020, August 19). *Tackling the small object problem in object detection.* Roboflow. https://blog.roboflow.com/detect-small-objects/

Soviany, P., & Ionescu, R. T. (2018) Optimizing the trade-off between single-stage and two-stage deep object detectors using image difficulty prediction. *2018 20th International Symposium on Symbolic and Numeric Algorithms for Scientific Computing (SYNASC),* (pp. 209-214). IEEE. https://doi.org/10.1109/SYNASC.2018.00041

Suzuki, K., (Ed.). (2013). *Artificial Neural Networks - Architectures and Applications.* IntechOpen. https://doi.org/10.5772/3409

The MathWorks, Inc. (2022). *Object detection using YOLOv3.* https://www.mathworks.com/help/vision/ug/object-detection-using-yolo-v3-deep-learning.html

Vahab, A., Naik, M. S., Raikarm P. G., & Prasad, S. R. (2019). Applications of object detection system. *International Research Journal of Engineering and Technology (IRJET), 6*(4), 4186-4192. https://www.irjet.net/archives/V6/i4/IRJET-V6I4920.pdf

Wang, R., Guo, Q., Lu, S., Zhang, C. (2019). Tire defect detection using fully convolutional network. *IEEE Access 7,* 43502-43510. https://doi.org/10.1109/ACCESS.2019.2908483

Weyssenhoff, A., Opala, M., Koziak, S., & Melnik, R. (2019). Characteristics and investigation of selected manufacturing defects of passenger car tires. *Transportation Procedia, 40*(2019), 119-126. https://doi.org/10.1016/j.trpro.2019.07.020

Wong, S. (2018, September 18). *21 casting defects and how to prevent them in your products.* Asia Quality Focus. https://www.intouch-quality.com/blog/21-casting-defects-and-how-to-prevent-them-in-your-products

Wu, Z., Jiao, C., Sun, J., & Chen, L. (2020). Tire defect detection based on Faster R-CNN. *Communications in Computer and Science Information 1336,* 203-218. https://doi.org/10.1007/978-981-33-4932-2_14

Yamashita, R., Nishio, M., Do, R. K. G., & Togashi, K. (2018). Convolutional neural networks: an overview and application in radiology. *Insights into Imaging*, *9*, 611-629. https://doi.org/10.1007/s13244-018-0639-9

Yao, S., Chen, Y., Tian, X., Juang, R., & Ma, S. (2020). An improved algorithm for detecting pneumonia based on YOLOv3. *Applied Sciences 10*(5). https://doi.org/10.3390/app10051818

Ying, X. (2019). An overview of overfitting and its solutions. *Journal of Physics: Conference Series 1168*(2). https://doi.org/10.1088/1742-6596/1168/2/022022

Zhou, I. (2021, March 18*). Data labeling of images for supervised learning.* Landing AI. https://landing.ai/data-labeling-of-images-for-supervised-learning/

APPENDIX

Missed Defect Annotations and Detections

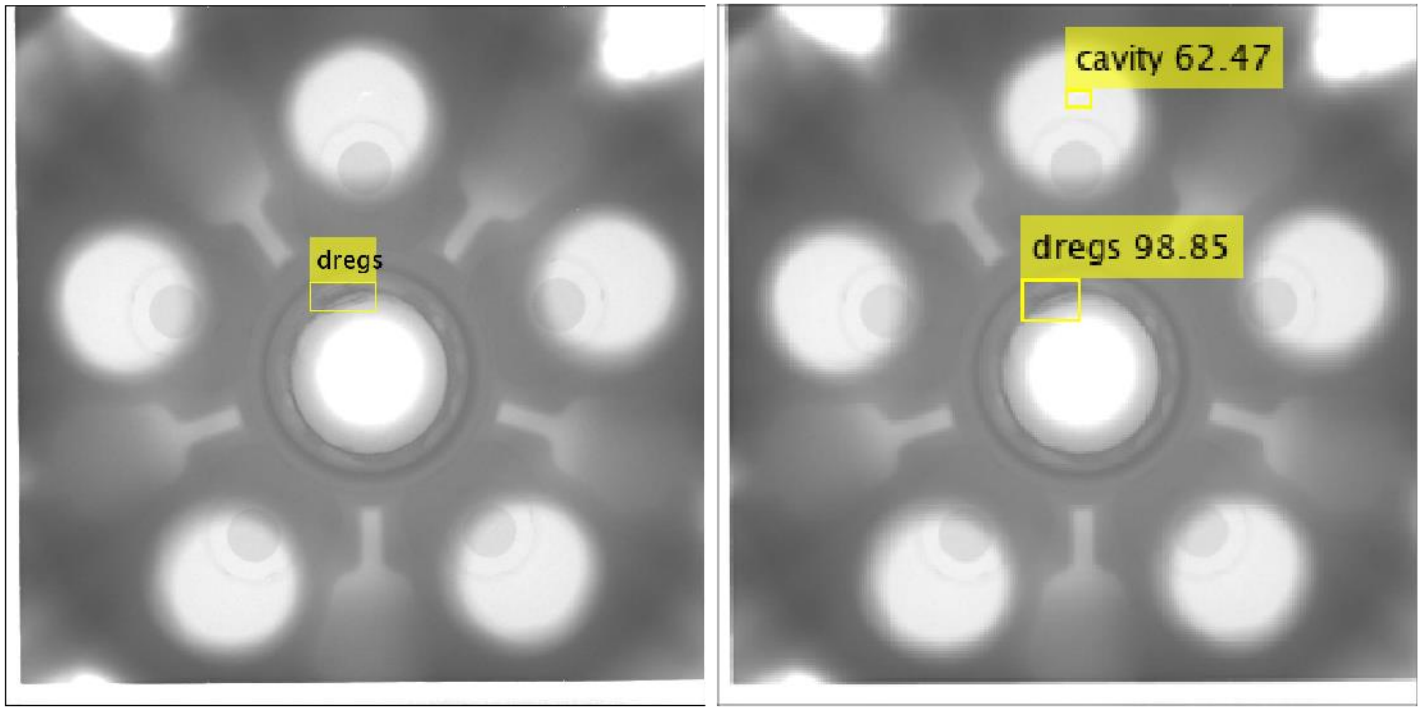

Figure A.1: Detection Results – Test Image 18

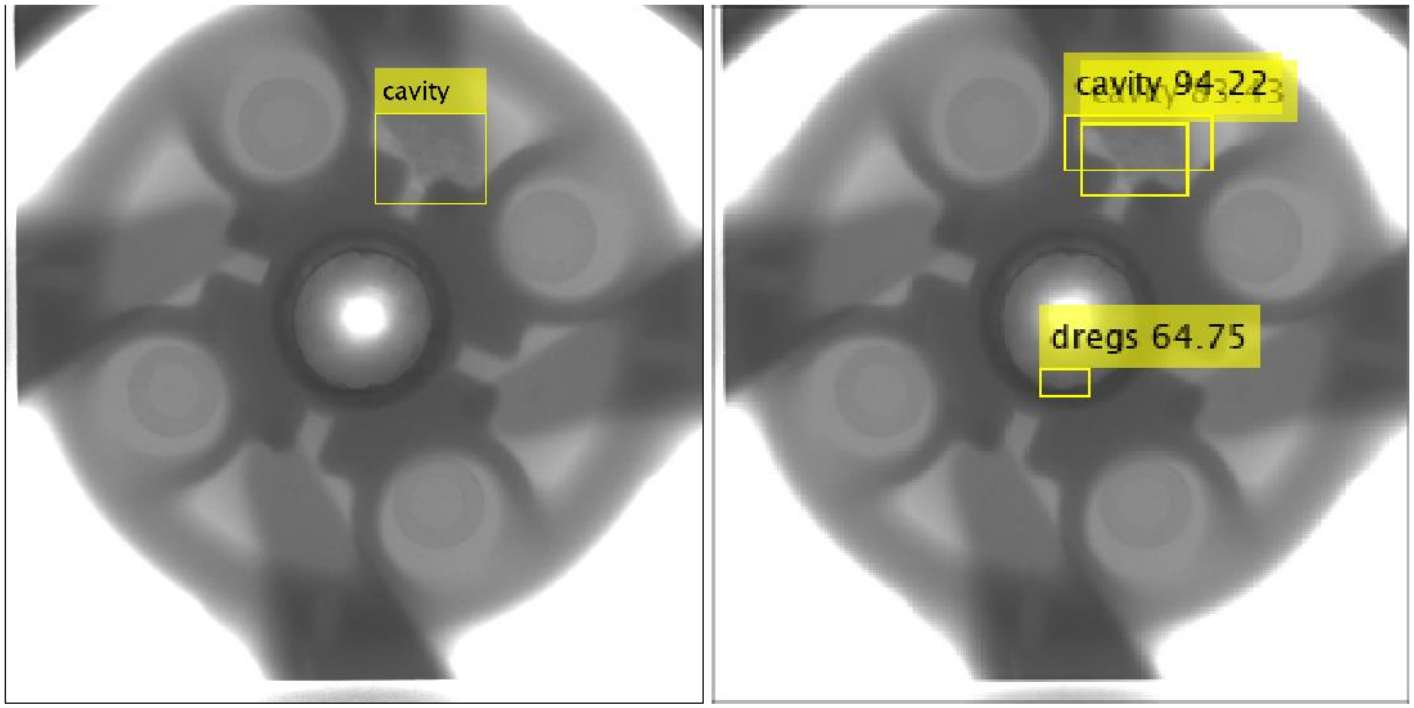

Figure A.2: Detection Results – Test Image 97

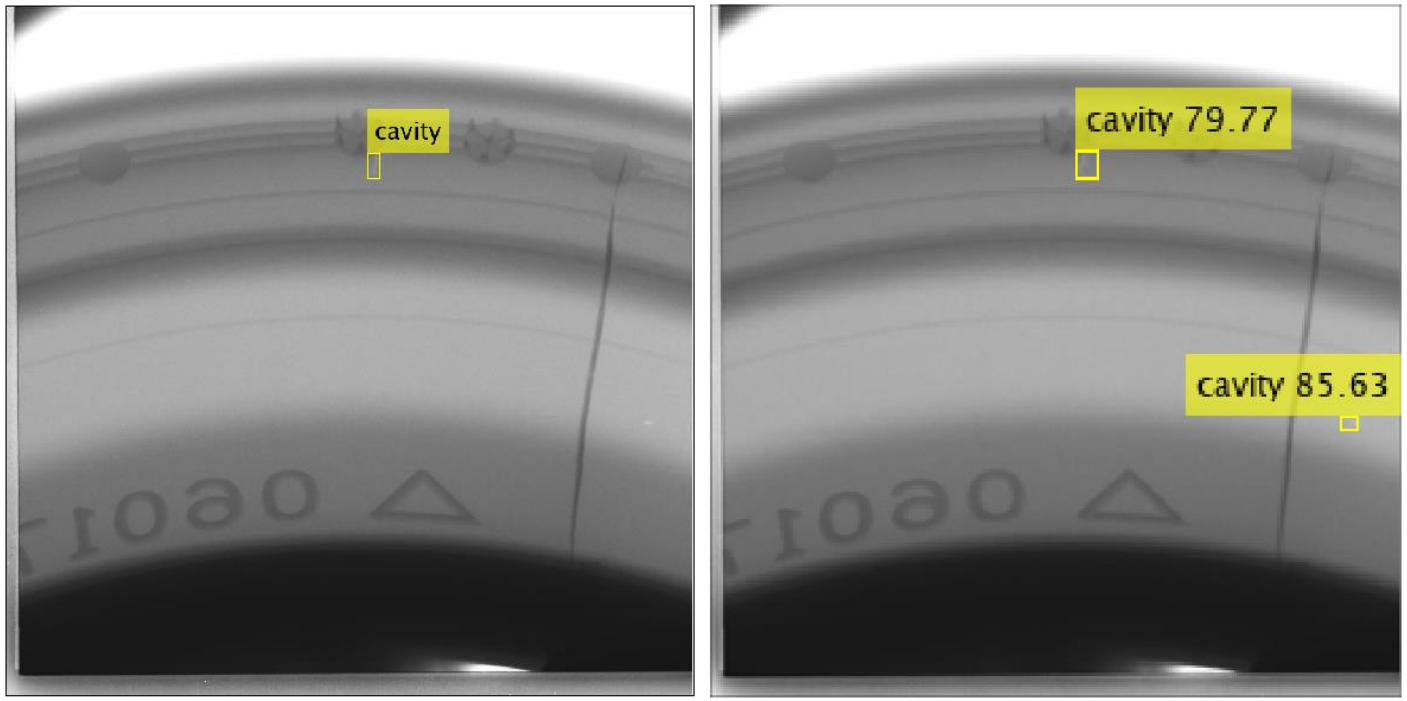

Figure A.3: Detection Results – Test Image 152

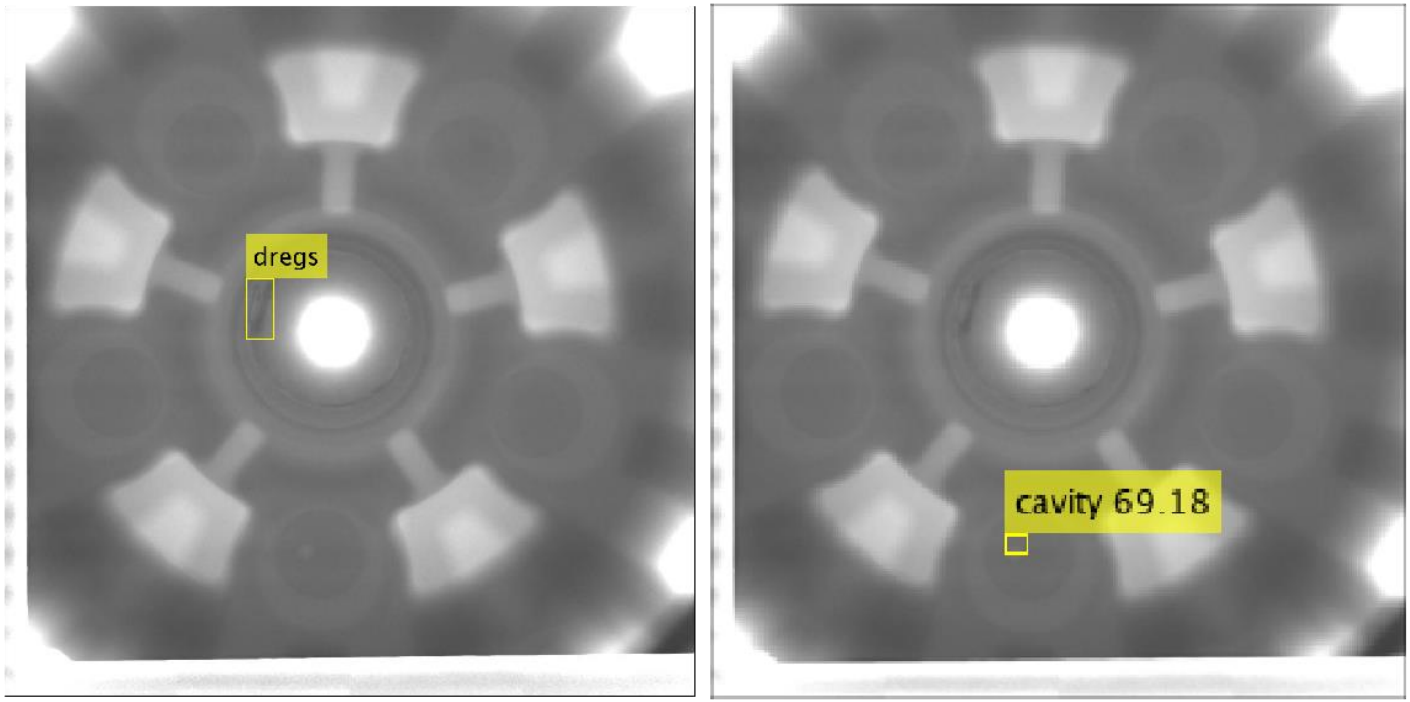

Figure A.4: Detection Results – Test Image 227

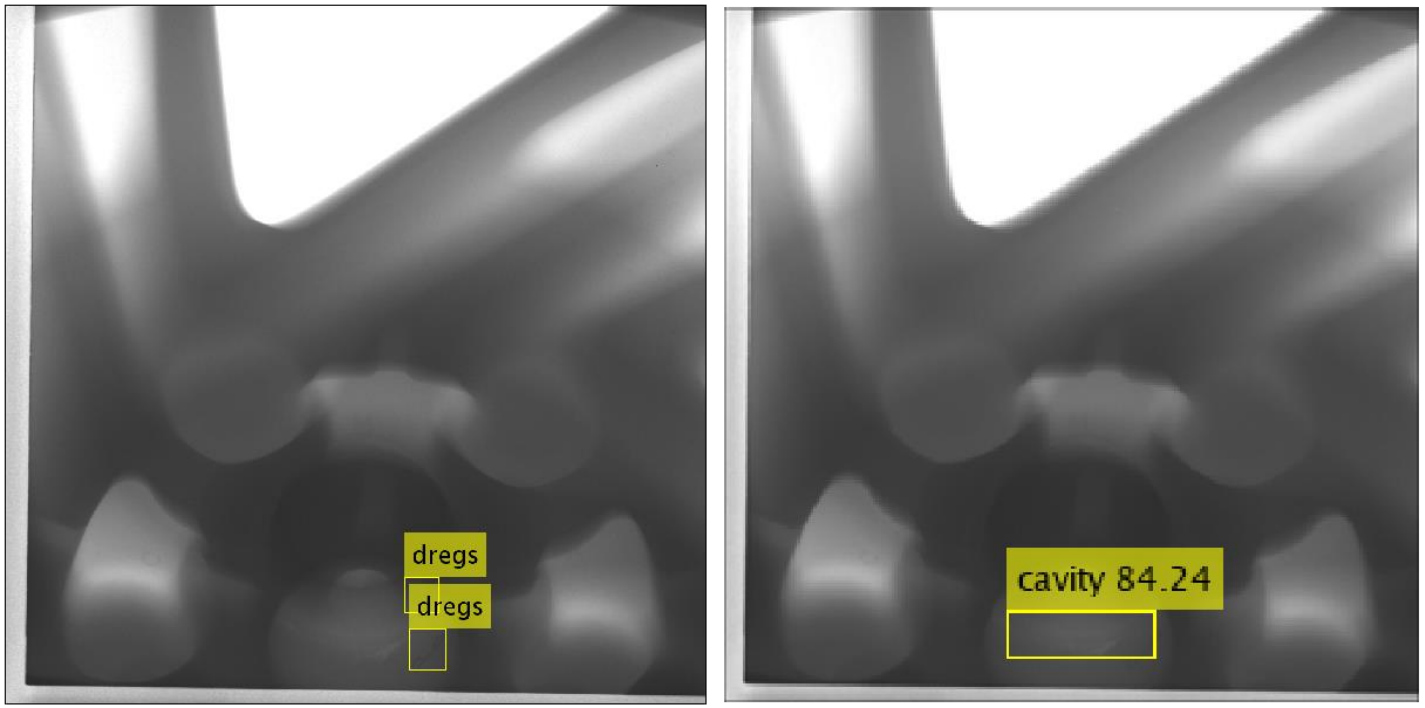

Figure A.5: Detection Results – Test Image 348

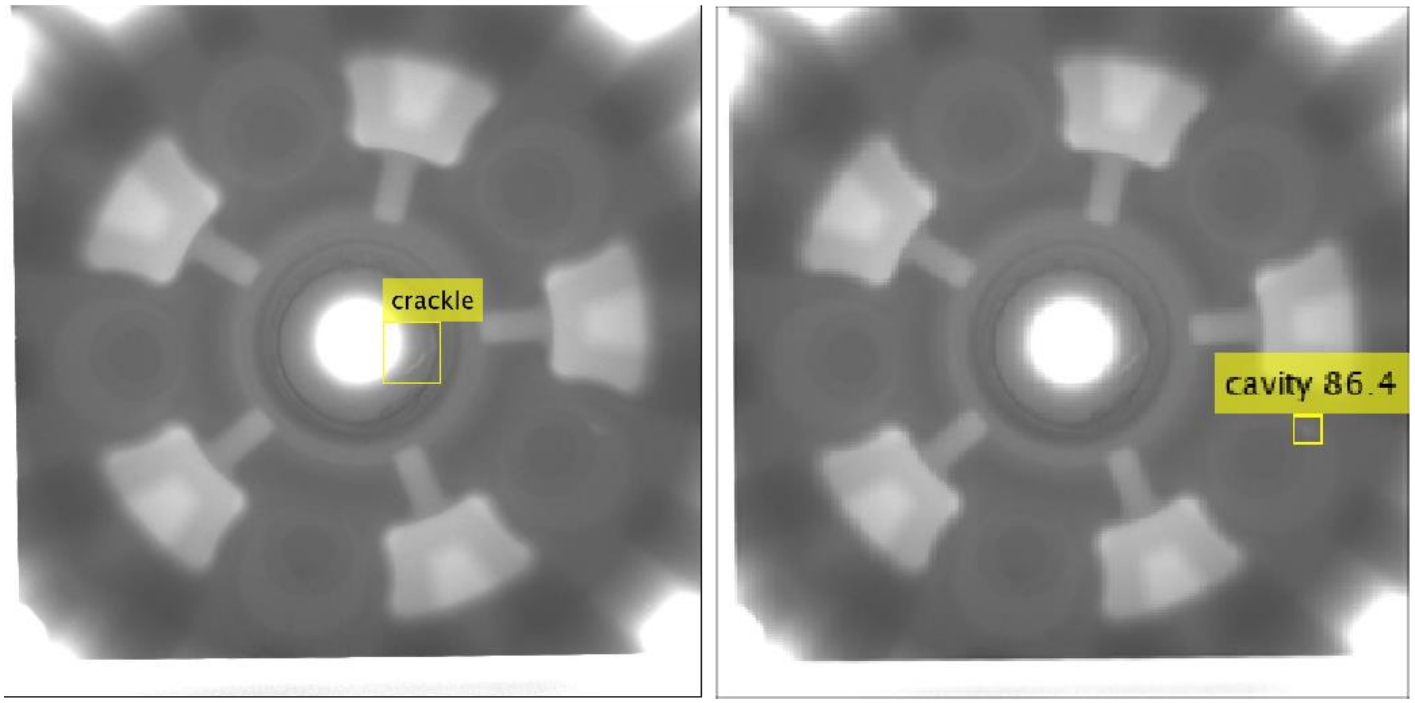

Figure A.6: Detection Results – Test Image 414