Building and Applying the Internet of Wasted Things: SCRAP and Periop Green

A Technical Report submitted to the Department of Computer Science

Presented to the Faculty of the School of Engineering and Applied Science
University of Virginia • Charlottesville, Virginia

In Partial Fulfillment of the Requirements for the Degree
Bachelor of Science, School of Engineering

Sonali Luthar
Spring, 2021

On my honor as a University Student, I have neither given nor received unauthorized aid on this assignment as defined by the Honor Guidelines for Thesis-Related Assignments

Signature  Date **5/15/21**
Sonali Luthar

Approved _____ Date _____
Madhur Behl, Department of Computer Science

Building and Applying the Internet of Wasted Things: SCRAP and Periop Green

Sonali Luthar
sl2ae@virginia.edu
University of Virginia
Charlottesville, Virginia

Abstract

Waste management is a difficult problem to solve. Unlike electricity or water management in building spaces, waste management has remained out of reach of Internet of Things (IoT) connectivity. This is why we develop the Internet of Wasted Things (IoWT): to bring intelligence to the overarching goal of monitoring, tracking, managing, and reducing waste. So far, we have explored two directions for application: SCRAP and Periop Green.

SCRAP is a closed-loop automated system which uses low cost cameras to automatically detect the item to be disposed and then guide the user towards the correct disposal bin in real-time. In doing so we constructed the Naturalistic Recycling Dataset (NRD), a novel data corpus which contains annotated videos of building occupants disposing commonly used waste items captured from multiple camera angles. Based on the diverse NRD dataset, we then created SCRAP, or the System for Classifying Recyclables through Automated Process, an automated waste disposal detection, tracking, and guidance system. SCRAP uses a customized transfer-learned deep neural network derived from the YOLOv3 object detector for real-time waste detection at the point of disposal. In our real world experimentation we show that SCRAP can achieve up to 93% mean average precision compared to baseline object detection networks. We implement and deploy SCRAP in a building environment in a closed-loop setting to prototype the system. SCRAP is motivated by the need to categorize waste at the time of disposal to prevent messy and manual sorting of recyclables downstream.

Periop Green tackles waste reduction in hospital operating rooms. During a surgery, most tools used by surgeons are single-use; that is, they are thrown away after one surgery. However, many of these tools are tossed out without ever being used. Along with the detrimental environmental implications of such vast amounts of waste, these items can cost upwards of thousands of dollars each. Therefore, the IoWT seeks to use computer vision to detect these single-use, sterile surgical supplies on operating room scrub tables. This will allow targeted suggestions for reducing costs and waste. Like SCRAP, our SUSS detector is trained used transfer-learning on a custom dataset derived from the YOLOv4 object detector. We show a greater than 96% mean average precision on our test data.

1 Introduction

The world is facing a recycling crisis that is burying cities and towns in tens of millions of tons of landfill waste each day [16]. As recycling costs skyrocket, more waste is ending up in landfills and incinerators. The problem of waste management became even worse when in 2018 China, the world's largest recyclable processor, stopped accepting most of the world's scrap plastic and cardboard due to contamination problems, and a glut of plastics overwhelming its own processing facilities [8]. China's 2018 national policy [2] banned the import of many types of recyclables that the U.S. previously exported. Prior to the ban, China had received and processed a nearly 45% of total global plastic waste since; the U.S. alone had been sending around 4,000 shipping containers full of recyclables to China per day [7]. Other countries have followed suit: Thailand temporarily banned plastic recyclables in June of 2018, followed by similar restrictions from Malaysia, Taiwan, Vietnam, and India. Waste contamination in the U.S. is high since recyclables are often dumped "single-stream" into one bin instead of multi-streamed or separated from the source. Now China has strict standards for recycling materials it accepts, requiring contamination levels in a plastic bale, for example, to be no more than 0.1%. The international markets have resulted in upheaval in the U.S. recycling industry, with municipalities responding to the loss of these markets by increasing the cost of recycling, shutting down programs, storing recyclables indefinitely, and incinerating landfill recyclables. Waste management is a difficult problem to solve. Individuals, especially in the United States, don't think much about where their waste will end up. However, unlike electricity or water management, waste management has remained out of reach of Internet of Things devices and connectivity. The ordeal is still messy, slow, and

, ,

ACM ISBN 978-x-xxxx-xxxx-x/YY/MM…$15.00
https://doi.org/10.1145/nnnnnnn.nnnnnnn

manual and often left to back-of-house processes. The development of the Internet of Wasted Things is an overarching project endeavor that seeks to modernize, automate, and increase efficiency of waste management. The umbrella has developed into two major avenues: first, the SCRAP model, and secondly, Periop Green.

The SCRAP Model

Waste management typically occurs downstream after commingled waste has been collected from buildings. While single-stream recycling is convenient, it makes the sorting process more cost- and time-prohibitive with less material being salvaged [18]. Proper categorization at the point of disposal itself would ensure cost-efficient and sustainable waste management [3].

In buildings, electricity and water management is now an integral part of real-time operations and planning. This has led to the development of new methods and tools to ingest such data and close-the-loop with energy and water management. However, tracking waste with the same ease has remained beyond reach. This thesis presents SCRAP, the System for Classifying Recyclables through Automated Process: a computer-vision enabled system for waste tracking in building spaces. This project is pertinent given UVA's 2030 Sustainability Plan which includes a goal to reduce UVA's waste footprint 70% by 2030 relative to 2010 levels. As UVA and other organizations seek to meet waste reduction goals, the research presented lays foundation for building **The Internet of Wasted Things (IoWT)**.

What is Recycling Contamination? Recycling contamination occurs when materials are sorted into the wrong recycling bin (for instance placing a glass bottle into a mixed paper recycling bin), or when materials are not properly cleaned, such as when food residue remains on a plastic yogurt container. Recycling these materials is sometimes referred to as *aspirational recycling*, as one is simply throwing material into the recycling bin with the hope that it will find its way to where it needs to be eventually. Unfortunately, this is rarely the case. When disposed of improperly or in the wrong bin, even recyclable materials like plastic and paper can act as contaminants.

Because costs of sorting recyclables adds quickly, even viable materials will often be sent to the landfill if contamination levels are too high [13].

Disposal mistakes are often caused by:

1. Confusing categories (often tried and failed to be solved by complicated posters),
2. Lack of knowledge of whether an item is recyclable,
3. Absent-mindedness or hurried mistakes, and
4. Lack of care, diligence, or discipline.

The pitfalls are many but the consequences of ocean plastics and climate change are very real. The problem is not just technical (how to sort the waste) but also human. Consequentially it requires an approach where we can leverage advances in computer vision and object detection to assist the humans in reducing contamination at the point of disposal. In doing so, we also strive to inculcate good recycling habits among users over time.

Periop Green

The United States healthcare industry wastes over $2 billion per day resulting in more than $750 billion wasted per year. This waste accounts for roughly 25% of healthcare expenditure [25]. This cost is hard to tackle due to the sensitive nature of the environment; it is difficult to justify potential cost savings at the expense of healthcare outcomes. For this reason, hospital administrators are constantly searching for ways to reduce dollars wasted. One such target area are hospital operating rooms. During surgeries, doctors have a variety of tools opened at their disposal for use. However, it is known anecdotally that the number of items at disposal is more than what is actually used during surgery. Because these tools are still open during the perioperative time, they must be either cleaned, or in the case of single-use items, thrown away, regardless of whether they were actually used. Some of these single-use items can range from single digits dollar amounts to thousands of dollars each. Because surgical rooms are isolated areas of work, documenting the unused waste is an almost impossible task—it requires coming in directly at the end of surgery and before the table is cleaned by nurses. To tackle this problem of perioperative waste, we look to the Internet of Wasted Things. With our system, we seek to use computer vision to detect which items are on the scrub table and when certain items leave the table during the surgery, indicating the item has been used.

Research Contributions

We present the following research contributions:

1. A new **Naturalistic Recycling Dataset (NRD)** which contains annotated videos of building occupants disposing a variety of everyday items. Such a dataset has not been explored for waste detection before.
2. The development of **SCRAP**, the System for Classifying Recyclables through Automated Process. SCRAP is trained on the NRD using transfer learning applied to a YOLOv3 model for the customized task of real-time waste item detection. To close-the-loop, upon identification of the item, visual cues are presented to the human to guide correct disposal to reduce recycling contamination (Figure 1).
3. Rigorous comparisons between different object detection approaches for training the neural network and demonstration of the working closed-loop SCRAP system.

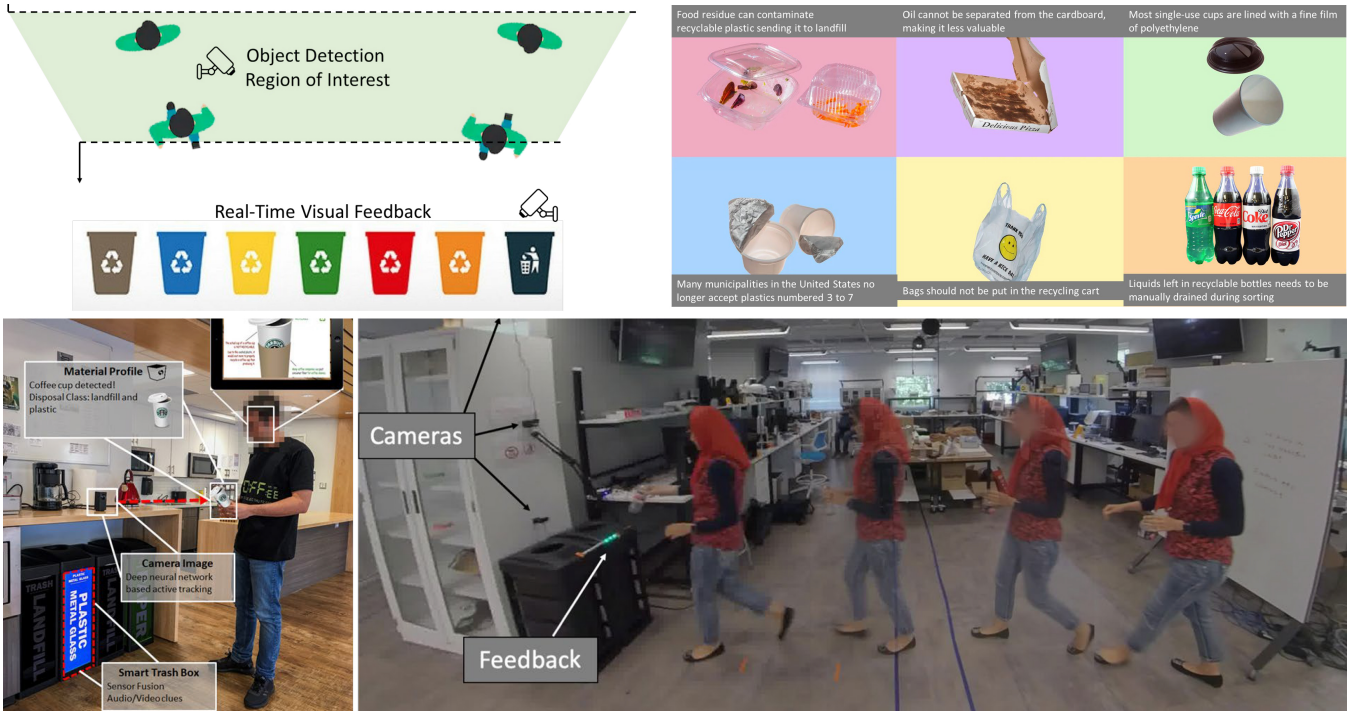

Figure 1. Top Left: Overview of the IoWT concept; Top Right: Examples of common recycling contamination; Bottom: IoWT automated detection and guidance setup and closed-loop implementation.

4. Creation of the **SUSS Detector** for detecting single-use surgical supplies, made as a proof-of-concept workflow for detection and classification of single-use tools on operating room scrub tables. This model is trained using transfer-learning on a custom SUSS dataset with a YOLOv4 base detector (Figure 18).
5. An exploration of adapting the detector results to identify single-use **sterile surgical supplies** (SUSSS), i.e. identifying which items on the scrub table have actually been used during surgery.

We believe both of these projects are well poised to being used in the real-world and making real-impact. Both project seeks to integrate IoT waste management into existing infrastructure to prevent hurdles that arise with the need for disruption. This is what makes our Internet of Wasted Things work so novel and impactful.

2 Related Work

Automated recycling has received a lot of attention from researchers in recent years. Authors have proposed interventions at all stages of the recycling pipeline, from waste detection and sorting at the building level, to large automated robots and machines for downstream sorting of waste. In this section, we present an overview of related work and compare it to the contributions of this paper.

2.1 Waste-related Datasets

TrashNet [27] is a dataset of over 2500 images of waste items in the categories: glass, paper, cardboard, plastic, metal, trash. The dataset is favorable for its breadth of recyclable material included; however, the images are only of the objects themselves against a plain white background as seen in Figure 2. Therefore, this dataset is undesirable for real-time, real-life detection of waste objects.

Similar to the TrashNet dataset is the TACO (Trash Annotations in Context) [19] dataset. TACO is a growing open source image dataset curated for the purposes of detecting litter in real life. Therefore, its images are of waste objects in the context of wildlife surroundings as found in the real world, as seen in Figure 2. While these could serve as a starting point to train waste specific object detectors, they lack the context of a human approaching a set of bins to dispose the item off. Therefore, it is not a sufficient dataset for our research.

2.2 Automated Waste Detection

A similar project to ours is the Garbage Detection with YOLO [9] which collects a variety of images of garbage across Google and then trains a YOLOv3 model to detect garbage bags, dumpsters, bins, and blobs. The goal of this research is to monitor and quantify garbage collection for many of the same reasons as our own research—that is, waste is an often ignored yet persistent environmental problem that

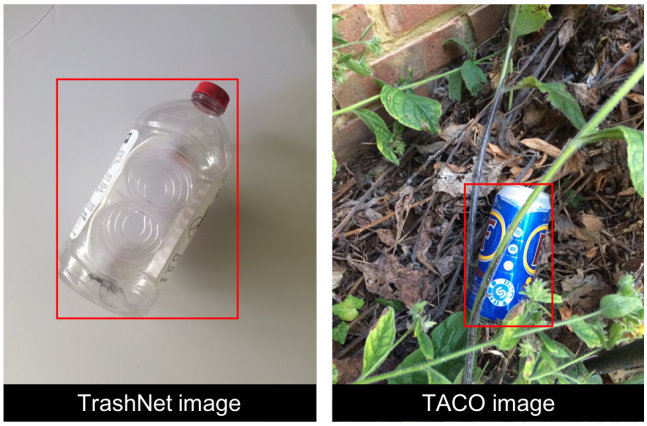

Figure 2. Existing waste datasets like TrashNet and Taco focus only on images of the object itself and do not contain any naturalistic recycling scenes.

currently eludes IoT management. While the Garbage Detection model achieves almost 60% mean average precision (mAP), it lacks in the exact area where our research excels. The system only monitors waste after it has already been disposed and collected at a garbage site, abstracting the IoT monitoring away from human interaction, i.e. "downstream" in the waste process. Our research advantageously seeks to monitor directly at the point of disposal before waste objects turn into unidentifiable garbage. By sorting, managing, and monitoring objects upstream, we are left with cleaner, more productive waste avenues in the recycling process. In addition, our research compares many training techniques for YOLOv3 to achieve an even higher final mAP than reported in this project.

Another related research is the Comparison of Deep Learning and Support Vector Machines for Autonomous Waste Sorting [24]. This paper compares the performance of CNN vs. SVM models in detecting waste objects. They use their own dataset which is again a set of images of various waste objects themselves, without surrounding context. The labels for the data are *plastic*, *paper*, or *metal*. This paper finds that SVM models perform better at an accuracy of 94.8% compared to the CNN accuracy 83%. This paper succeeds in its rigorous comparison of model performance but its findings fail to translate to real-world applications, again due to the fact that objects are not detected in real-world context. As such, we have decided to train with a CNN model in our own research. This is because CNNs are generally better at using surrounding context in the image in its object detection and we believe that context is more applicable for our purposes. This research also neglects to imagine the application of their developed model; it is unclear at what point in the waste stream a model like this could be applied for better waste management. Our research therefore ensures to clearly define the placement of our model in the waste disposal pipeline in order to productively monitor, manage, and sort recyclables.

2.3 Downstream Recycling and Sorting

Recycling "robots" are one form of commercial solution to sort materials downstream. One example is Max-AI [26]. This robot identifies recyclables in a Material Recovery Facility using multi-layered neural networks and a vision system. Max-AI seeks to solve the issue of recycling after it has been disposed and collected. This may be advantageous in some ways because it prevents humans from getting involved with the technology, and of course humans are prone to increasing complexity and confusion. But ultimately, downstream sorting is the messiest point in the waste disposal stream to intervene. At this point, much of the recyclable material has already been contaminated and is either impossible or difficult to salvage. Therefore, it is more ideal to sort waste upstream, i.e. with human intervention, as our project seeks to do. Additionally, robot sorters are not scalable—they are expensive, large pieces of technology that must be adopted at recovery facilities in order to make meaningful change. In contrast, our SCRAP idea is lightweight and portable; it can be attached to any bin location for instant aid in sorting recyclables.

ZenRobotics ([28]) is another company who produces similar robot waste sorters. ZenRobotics has two sorters, Heavy Picker and Fast Picker, each designed for a different type of waste: the former being used for commercial and industrial waste, construction waste, plastic, and scrap metal, and the latter being used for municipal solid waste and packaging. The robots are designed with an AI network called ZenBrain. These sorters have the advantage of being specialized for different types of waste. They are also tackling a different waste set than ours; our SCRAP system is not built to handle industrial or construction waste—for these cases a robot sorter is likely the better solution. However, the ZenRobotics sorters have a similar issue of being expensive and hard to scale, and they are still forced to sort waste downstream rather than upstream.

3 Naturalistic Recycling Dataset

A major bottleneck for building a waste detection system is the lack of naturalistic datasets. Most waste-related datasets only contain images of the items themselves, without any context of how they are/will be disposed. The ideal dataset for our purposes is that which includes images and/or videos of many individuals walking forward towards recycling bins with a variety of waste objects in hand. This is because the goal of our object detector is to identify recyclable objects just as they are being placed into the bins, and so our data should reflect the environment in which our model would be deployed. In the real world, when someone approaches a set of bins to dispose the object, there are various factors which

come into play while detecting the object. For instance, the object can be momentarily occluded behind the person's body, the position of the camera may detect different angles of the object, etc.

Therefore to build our proposed system, we began by creating a first-of-its-kind dataset which captures the naturalistic actions of the humans as they dispose everyday items in an office building environment. We call this Naturalistic Recycling Dataset—it contains not a library of images of waste items, but instead contiguous videos and image sequences of building occupants disposing these items. In this section, we provide an overview of the NRD data and how it was collected and organized for the specific purpose of automated waste detection in the real world.

3.1 Data Collection

The collection of the data has its own set of challenges. These include: determining the ideal position of the camera relative to the waste bins, choosing which camera is best for our purposes (small, cheap, and scalable), and selecting a set of commonly disposed waste items W to include in the dataset.

In order to determine the ideal camera placement, we conducted trials with cameras positioned at three different heights above the ground: 3 feet (bin level), 5 feet, and 7 feet (overhead) as shown in Figure 3. To accurately evaluate the performance at each height, we varied the distances of the participants away from the trash bins as well at 3 feet, 5 feet, 10 feet, and 15 feet. At each of the distances, for each of the cameras, we used the baseline YOLOv3 object detector for bottle detection as the metric for evaluating the camera placement performance. Overall, the overhead performed the best at detection over all distances (due to a wider field of view), and the camera at the bin-level position performed the second best. Therefore, for our naturalistic data collection, we chose to place cameras at those two positions, as seen in Figure 4.

Next, we chose a list of waste objects to include in our dataset. We selected common waste items that are often recycled incorrectly. In the NRD data, *bottle* is one of these categories. In general, plastic bottles are an "easy" recyclable

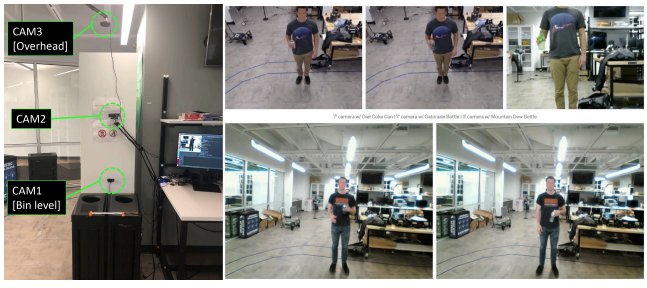

Figure 3. High and low camera positions during data collection.

item; however, even they may be destined for the landfill if they are contaminated with liquid. The category *bottle* is also one of the categories present in the YOLOv3 detector, and so including it in NRD allows for bench-marking against YOLOv3. The next item category in the dataset is *cup*. Coffee cups are all too common, especially in work and university settings, but they vary widely in how to be recycled. Most commonly, the lid of the coffee cup falls into plastic, the cardboard sleeve falls into paper, and the waxed-line cup itself is destined for the landfill. Proving our ability to detect cup objects as whole in this research sets the stage for detecting it in pieces in the future. Lastly we chose to use a variety of *paper* items in our dataset. This was to tackle the heterogeneous nature of waste items; although paper can take many forms and likely look very different to a computer vision model, they are often all still recyclable under the category *paper*. So in summary, the set of waste items in NRD are comprised of a variety of *bottle*, *cup*, and *paper* objects.

The data was collected by having building occupants walk up to the bin location and disposing the waste items while being simultaneously recorded from two camera locations. The identity and faces of individuals is anonymized in the dataset. Overall the dataset is comprised of 7 total videos. Each video contains dozens of example clips of several occupants disposing one item at a time (between bottles, paper, and cups) into the bin location. The natural behavior, motion, and pose of the occupants gives this dataset its unique properties. NRD also contains examples of occupants walking across the field of view of the cameras and not disposing any items at all.

3.2 Data Annotation

The Computer Vision Annotation Tool (CVAT) was used for adding ground truth annotations to the NRD data. CVAT is an OpenCV project made for labeling for computer vision datasets. CVAT provides an easy-to-use interface to make object detection annotations simple and efficient. In the NRD, we have three class labels: Bottle, Cup, and Paper. In object detection, we usually use a bounding box to describe the target location of the object in an image. The bounding box is a rectangular box that can be determined by the x and y axis coordinates in the upper-left corner and the x and y axis coordinates in the lower-right corner of the rectangle. Using CVAT, one can draw a box around the object of interest for the annotation. Then we can assign the true object class, among the defined class labels, to the bounding box. The tool supports the ability to interpolate bounding box annotations across multiple video frames and the ability to mark attributes for any labeled object.

Using CVAT, we can then export the ground truth annotations for each image sequence in NRD in a format compatible with the widely-used YOLO object detection. For each `.jpg` image-file in the data, a `.txt` file with the same name is created. The `.txt` file contains the class label number and

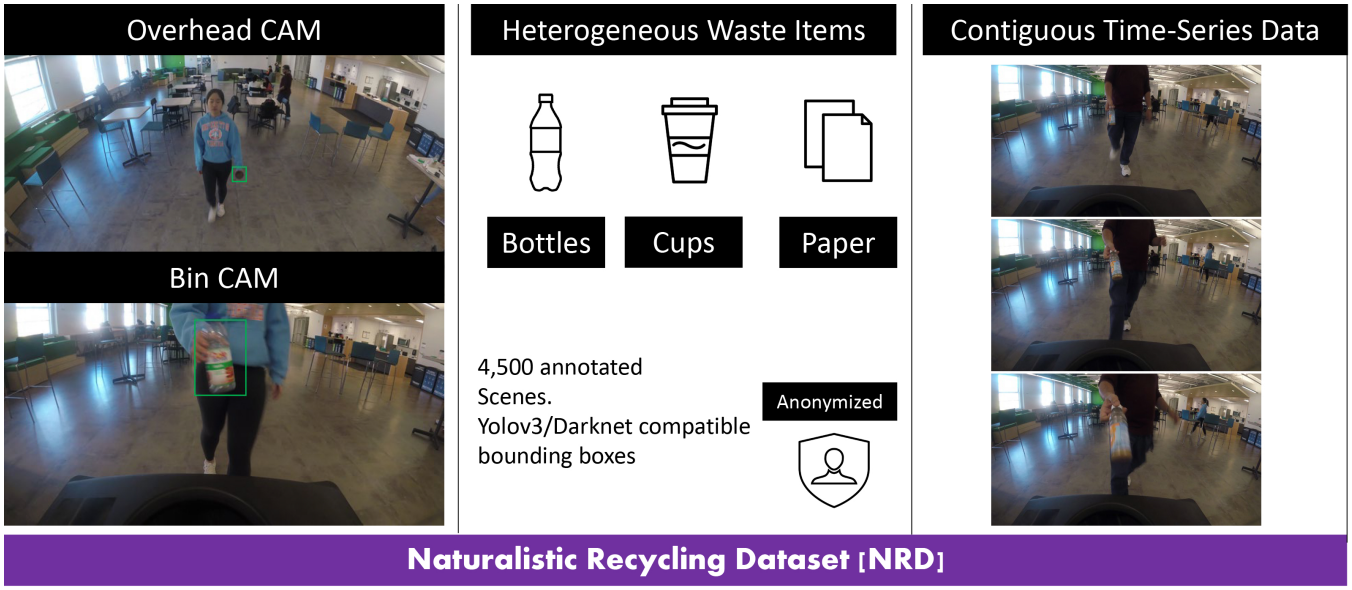

Figure 4. Overview of the novel Naturalistic Recycling Dataset [NRD]. The first-of-its-kind dataset contains natural scenes of occupants disposing three different object types—Bottles, Cups, and Paper. The dataset contains images from multiple camera positions recording above a waste-bin location in an office building. The data is contiguous in time and allows for a richer tracking of waste items. The data is anonymized and no personal identifiable information is recorded.

bounding box coordinates for the object(s) in this image. For each image we annotated, we only had one object of interest. For each object, the annotation format is:

$$< object - class >< x >< y >< width >< height >$$

Where:

<object-class> is an integer number of object from 0 to (classes-1) and <x> <y> <width> <height> are float values relative to width and height of the image.

3.3 Image Augmentation

Augmentation of an image dataset is the process of making minor edits, such as changing blur, contrast, lighting, etc., to each image i in the dataset; this creates a larger, more diverse image dataset without the need for collecting or annotating more data. We used the augmentation library *imgaug* [12], a python library designed for image augmentation in machine learning applications. We chose this library for its simplicity and ease-of-use, as well as the variety of augmentation options provided. Each image was augmented by varying its brightness, blur, hue, and temperature. These techniques alter the look of the image and aid the object detection model for use in new environments due to differences in background, lighting, colors, etc. We chose to use the following augmentation techniques in our dataset:

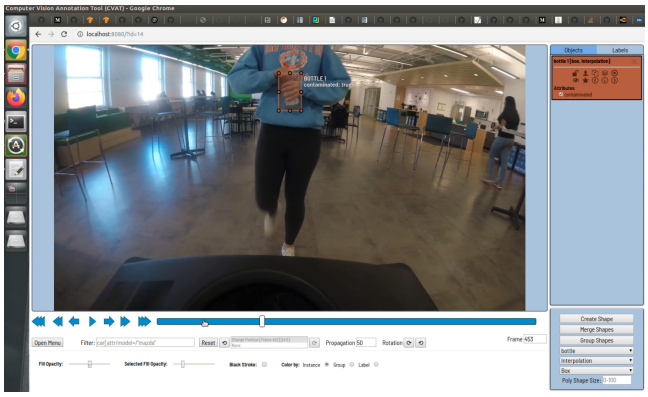

Figure 5. CVAT tool used for image annotation for NRD.

1. **Brightness**: The image is converted to a random HSV color-space, the brightness-related V channel is augmented by adding a random number from -50 to 50 which is then integrated back into the image, and the image is then converted back into the original color-space.
2. **Blur**: A Gaussian blur is applied to the image to with sigma=2.5.
3. **Hue**: The hue (H) channel in the image's original HSV color-space is extracted and added to a random value between 0 and 50, then integrated back into the image.
4. **Temperature**: The temperature of the image is changed to a random Kelvin value between 1,100 and 10,000 where low values are warm tones and high values are cool tones.

An example image and its augmentations are shown in Figure 6.

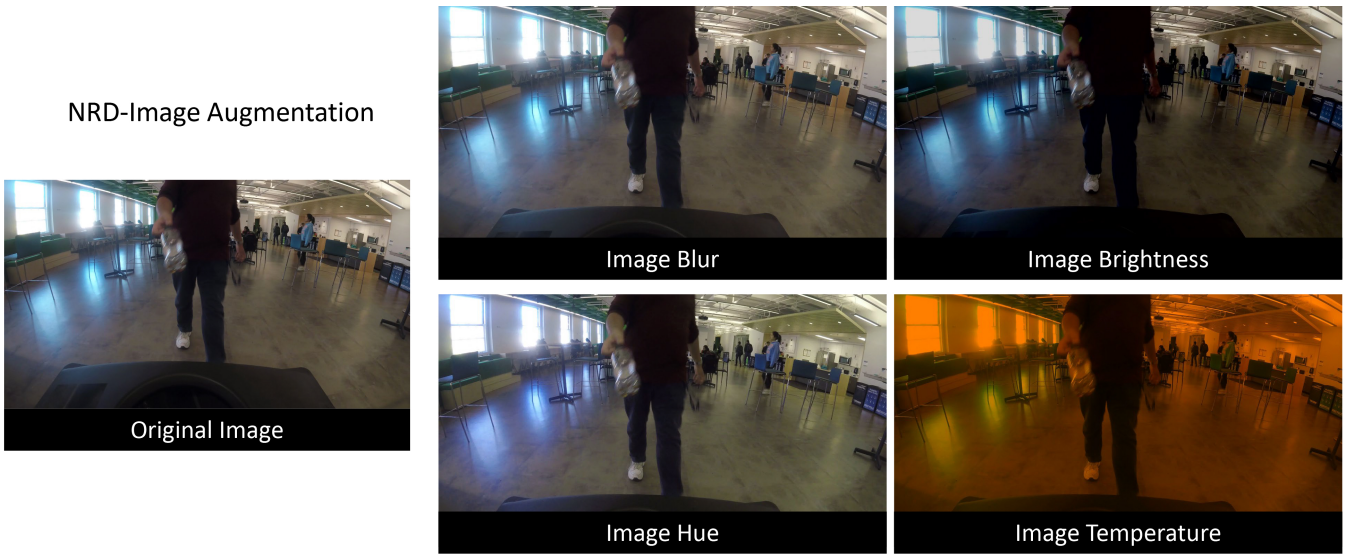

Figure 6. Examples of image augmentation techniques applied to the NRD dataset.

3.4 Dataset Organization

NRD includes a total of 4,491 annotated images. These include: 3,472 *bottle* images, 278 *cup* images, and 741 *paper* images, all derived from the naturalistic waste items disposal scene. To train our custom waste items detector, we randomly select 80% of the images for training and use the remaining 20% for testing.

Image augmentation added an additional $4,491 * 4 = 17,964$ augmented images to the dataset. With the inclusion of the original images, NRD now has a total of 22,455 images.

The NRD dataset goes beyond other existing datsets to aid in waste detection and classification. By using naturalistic video data converted to still images, we have embedded temporal and spatial properties of how a waste object moves towards a recycling bin into our data. The contiguous image sequences could aid with identifying human behavior and patterns with certain waste objects and/or detecting the individual's intention or lack thereof to dispose of the object. The dataset also has examples of occlusion when the waste object is hidden behind a person's body or mostly hidden within their hand. This could be leveraged to track object permanence even if it momentarily disappears from the field of view of the camera(s).

Even though the NRD dataset is unique and innovative, it does have a few limitations. The image sequences were recorded with the same ambient background and under uniform, well-lit conditions. Additionally, the types of objects used in the dataset are a subset of all of the types of objects that could be thrown away. In particular, we chose to focus on plastic bottles, coffee cups, and various paper products because they are common and typical waste objects; however, we can imagine that this limits the potential for recycling detection, such as with aluminum foil, organic waste, snack wrappers, etc. Nonetheless, the NRD dataset is extremely useful for training a computer vision-based custom waste items detector. The authors will release the anonymized dataset in the public domain.

4 SCRAP - System for Classifying Recyclables through Automated Process

Creating the NRD dataset allows us to develop an automated waste detector guidance system which we call SCRAP. We first describe the YOLOv3 detection framework, the backbone of our model. Then we describe the creation of our novel waste object detector model that is used for waste detection in SCRAP.

4.1 YOLOv3 Object Detection Background

We chose to use YOLOv3 (You only look once) [23] for our computer-vision detection framework to serve as the backbone of our project. YOLO is a state-of-the-art, real-time object detection system and faster than other comparable detectors. It has comparable mAP (Section 5.1) to RetinaNet [14], another high-accuracy object detector, but is about 4x faster. Because of our need to perform object detection in real-time, we chose YOLO as a detection framework for its speed. YOLO processes an entire image with a single neural network; this is in contrast to other detectors which apply a classifier at multiple locations in an image and mark high-scoring regions as detections.

There are a variety of improvements made to YOLOv1 [21] and YOLOv2 [22] to create YOLOv3. Most notably is the new feature extractor named Darknet-53 [20]. This network has 53 alternating 3x3 and 1x1 convolutional layers, an increase from the 19 convolutional layers in the feature extractor of YOLOv2. Darknet-53 is also interspersed with shortcut

layers, as well as an average pool and softmax layer at the end for classification, for a total of 78 layers in its network. Darknet-53 is trained for classification on the ImageNet [10] dataset.

The YOLOv3 detector model is created by removing the final three output layers of Darknet-53 (including the final convolutional layer), i.e. preserving the first 75 convolutional and shortcut layers of Darknet-53; then, appending 23 more convolutional layers, along with several route, upsample, and yolo (detection) layers. An abbreviated version of all of the layers in the YOLOv3 architecture can be seen in Figure 7. This results in a total of 107 layers in the YOLOv3 object detector model. There are three yolo layers meaning detection is performed at three scales, and each prediction improves upon the prior computation to extract more fine-grained features from the image. The a visualization of the entire YOLOv3 architecture is shown in Figure 8. When training the YOLOv3 object detector, the Darknet-53 ImageNet classification weights are used for initialization, with the remaining layers beginning at random weight values. The YOLOv3 baseline model is trained on the COCO (Common Objects in Context) dataset [15]. The model includes 80 classes for detection, such as *dog*, *chair*, *person*, etc. It also includes 2 waste related categories - bottles and cups. YOLOv3 has a mAP of 57.9% on the COCO test-dev benchmark, the default COCO test dataset consisting of about 20K images.

```
layer     filters    size              input                output
    0 conv     32  3 x 3 / 1   416 x 416 x   3   ->   416 x 416 x  32
    1 conv     64  3 x 3 / 2   416 x 416 x  32   ->   208 x 208 x  64
    2 conv     32  1 x 1 / 1   208 x 208 x  64   ->   208 x 208 x  32
    3 conv     64  3 x 3 / 1   208 x 208 x  32   ->   208 x 208 x  64
    4 Shortcut Layer: 1
    5 conv    128  3 x 3 / 2   208 x 208 x  64   ->   104 x 104 x 128
    6 conv     64  1 x 1 / 1   104 x 104 x 128   ->   104 x 104 x  64
    7 conv    128  3 x 3 / 1   104 x 104 x  64   ->   104 x 104 x 128
    8 Shortcut Layer: 5
        .
        .
        .
   71 Shortcut Layer: 68
   72 conv    512  1 x 1 / 1    13 x  13 x1024   ->    13 x  13 x 512
   73 conv   1024  3 x 3 / 1    13 x  13 x 512   ->    13 x  13 x1024
   74 Shortcut Layer: 71
-------------------------------------------------------------------------
   75 conv    512  1 x 1 / 1    13 x  13 x1024   ->    13 x  13 x 512
        .
        .
        .
   81 conv     18  1 x 1 / 1    13 x  13 x1024   ->    13 x  13 x  18
-------------------------------------------------------------------------
   82 detection
   83 route  79
   84 conv    256  1 x 1 / 1    13 x  13 x 512   ->    13 x  13 x 256
   85 upsample           2x    13 x  13 x 256   ->    26 x  26 x 256
   86 route  85 61
   87 conv    256  1 x 1 / 1    26 x  26 x 768   ->    26 x  26 x 256
        .
        .
        .
   93 conv     18  1 x 1 / 1    26 x  26 x 512   ->    26 x  26 x  18
   94 detection
   95 route  91
   96 conv    128  1 x 1 / 1    26 x  26 x 256   ->    26 x  26 x 128
   97 upsample           2x    26 x  26 x 128   ->    52 x  52 x 128
   98 route  97 36
   99 conv    128  1 x 1 / 1    52 x  52 x 384   ->    52 x  52 x 128
        .
        .
        .
  105 conv     18  1 x 1 / 1    52 x  52 x 256   ->    52 x  52 x  18
  106 detection
```

Figure 7. An abbreviated description of all 107 layers of the YOLOv3 architecture. The tuning of these layers for our transfer-learned model, Model 2, is depicted in Figure 8. The dotted lines indicate separation between frozen, pre-initialized, and random weights.

YOLOv3 was trained using the following hyper-parameters:

- Batch = 64
- Subdivisions = 16
- Image width = 416
- Image height = 416
- Momentum = 0.9
- Weight decay = 0.0005
- Learning rate = 0.001
- Number of anchor boxes = 9 (3 per scale)
- Anchor boxes = (10,13), (16,30), (33,23), (30,61), (62,45), (59,119), (116,90), (156,198), (373, 326)
- Number of batch iterations = 500,200

Additionally, each convolutional layer preceding each of the three yolo detection layers had a filter of 255, equal to (num_classes + 5) * 3.

4.2 SCRAP: NRD-Trained Object Detector

We used a pre-trained off-the-shelf YOLOv3 on the NRD test data (see Table 1) as a baseline comparison but found that it was completely unfit for our application. The model was not developed to detect our types of waste objects in a building environment setting—notably, the category *paper* does not even exist in YOLOv3's 80 categories.

Consequentially, we set out to train our own custom waste object detectors trained on the NRD dataset. In particular we implement two methods - train the entire YOLOv3 network, and a transfer-learning method using YOLOv3 as the backbone architecture. These techniques allowed us to develop Model 1 and Model 2, respectively. In addition, for each of these networks we trained with and without the augmented image set to study the effect of image augmentation.

Before training, we compiled Darknet with CUDA [17] and OpenCV [6]. Integrating OpenCV with Darknet allows for support of most image file formats and real-time detection on a webcam.

4.2.1 SCRAP: Model 1 - Retrained YOLOv3. The training approach used for Model 1 is the typical mode of training any YOLOv3 network from scratch. The feature extractor Darknet-53 was used for initialization weights for the first three quarters of the architecture layers, with the last quarter being initialized at random weight values. Then, the network was trained using the NRD dataset instead of the COCO dataset. This training approach was time consuming due to the fact that all layers required re-tuning, in total taking around 16 hours for completion.

The number of categories for Model 1 were reduced from 80 in the baseline to 3 for our purposes (*cup*, *bottle*, *paper*). The hyper-parameters for Model 1 were the same as that

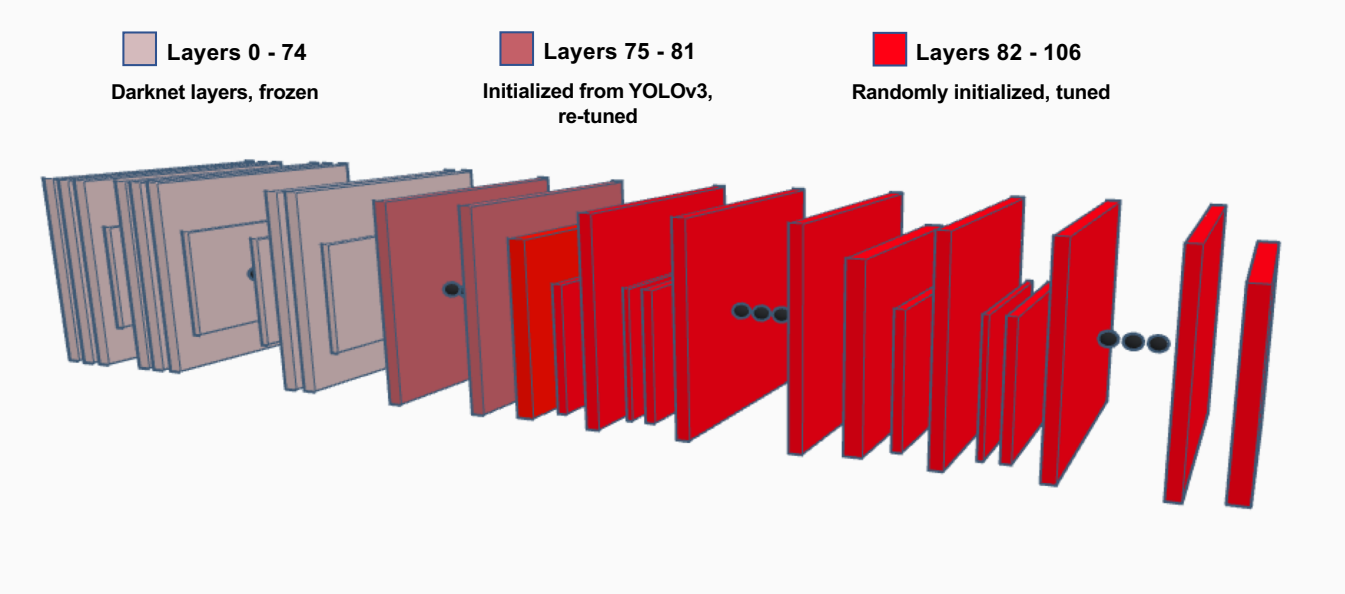

Figure 8. Network architecture for SCRAP-Model 2 object detector, transfer-learned from the Yolov3 generic object detector by freezing Darknet-53 layers and re-tuning the deeper classification layers on the NRD dataset. The layers shown correspond to the layers defined in Figure 7.

for YOLOv3 except for the number of batch iterations which was adjusted to 6000. Additionally, the number of filters was decreased to 24 for the convolutional layer preceding each yolo layer.

4.2.2 SCRAP Model 2 - Transfer Learning with YOLOv3.

For Model 2 we trained a new network using a technique known as transfer learning. Transfer learning allows a model to retain knowledge from previous training on other datasets that are applicable to the current use. Datasets such as COCO on which YOLOv3 is trained contains thousands of images. For instance, ImageNet has over one million labeled images. However, we often do not have so much labeled data in other domains (such as waste items in our case). Therefore, training deep learning models for object detection on small datasets can lead to severe over-fitting. Transfer learning is a technique that addresses this problem. The idea is simple: we can start training with a pre-trained model, instead of starting from scratch. This is desirable for a number of reasons, not least: (a) The backbone models (such as YOLOv3) have learned how to detect generic features from photographs, given that they were trained on more than hundreds of thousands of images for dozens or hundreds of categories; (b) The pre-trained models achieved state of the art performance and remain effective on the specific image recognition task for which they were developed; and (c) The model weights are provided as free downloadable files and many libraries provide convenient APIs to download and use the models directly.

In our case, we would like to retain the learned properties from the ImageNet and COCO datasets to use in our waste object detection. We do this by keeping the weights of those layers constant, or "frozen", during training, instead of allowing them to be re-tuned under the new data. [4] discusses the advantages of transfer learning, and from that we deem transfer learning is a useful technique for our purposes because:

1. Spatial patterns within images, as related to typical human environments, can be learned from any set of images, not just our custom images. These patterns may include the shapes, shadowing, curvatures, depths, etc. that are seen in images.
2. Using transfer learning allows a model to develop robust custom detection without requiring an extremely large image dataset, as was the case for us.
3. Transfer learning increases the likelihood that a model will perform well when placed in other environments because the first layers are trained on an entirely different dataset than the last layers.

The full architecture of our transfer-learned Model 2 can be seen in Figure 8. To implement transfer learning, we first extracted the weights of the first 81 layers of the baseline YOLOv3 model. These are the gray- and brick-colored layers as shown in Figure 8. Then, we used these weights for initialization while training (as opposed to only using the 75-layer weights from Darknet for initialization like in Model 1). Lastly, we froze the Darknet layers of the architecture, i.e. the first 75 layers of the model (shown in gray in Figure 8.

Weights in layers 82 through 107 were randomly initialized and fully trained, shown in red in the diagram. With this setup, we preserved the pre-trained weights from YOLOv3 into our own model, thereby preserving features learned from both the ImageNet and COCO datasets. This method of training was much faster because only a quarter of the layers needed tuning. Training for Model 2 took approximately 5 hours (compared to the 16 hours for Model 1). The hyper-parameters used for Model 2 were the same as those used for Model 1.

5 SCRAP Results and Deployment

The goal of our experiments is to rigorously compare Model 1 and Model 2 (see Section 4.2) to each other and against the baseline YOLOv3 model. Doing so allows us to:

1. Quantify the waste detection performance for each model on the NRD test data.
2. Evaluate the effects of data augmentation on the network's performance.
3. Identify the best candidate neural network for use in SCRAP system deployment.

5.1 Performance Metrics

We present an overview of the evaluation metrics used for the comparison between the YOLOv3 baseline model and our Models 1 and 2. Object detection models are evaluated mainly through the metric of mean average precision (mAP) [11]. Before defining mAP we must understand a key object-detection metric, the intersection over union (IoU). For detectors that output a correctly-labeled bounding box, the accuracy of the detection is measured through assessing the area of overlap with the ground truth bounding box. Let the coordinates of the ground truth bounding box (as obtained form the NRD data) be $(x1_g, y1_g, x2_g, y2_g)$ where $(x1, y1)$ are the coordinates for the top left corner of the box, and $(x2, y2)$ are for the bottom right corner. Similarly, let the coordinates of the predicted bounding box be $(x1_p, y1_p, x2_p, y2_p)$. Then, the IoU calculates the area of intersection of over the area of union between the two bounding boxes. That is,

$$intersectionBox = (x1_i, y1_i, x2_i, y2_i)$$

$$intersectionArea = (x2_i - x1_i) * (y2_i - y1_i)$$

$$unionArea = ((x2_g - x1_g) * (y2_g - y1_g)) + ((x2_p - x1_p) * (y2_p - y1_p)) - intersectionArea$$

$$IoU = \frac{intersectionArea}{unionArea}$$

Depiction of the bounding boxes and their areas of union and intersection are illustrated clearly in Figure 9.

IoU is useful in determining the "correctness" of a bounding box prediction. Typically a predicted bounding box is considered "correct" if it has an IoU of at least 0.5. This is the number that we also use in our evaluation, however it is possible to vary to IoU threshold such as to 0.25 or 0.75 and

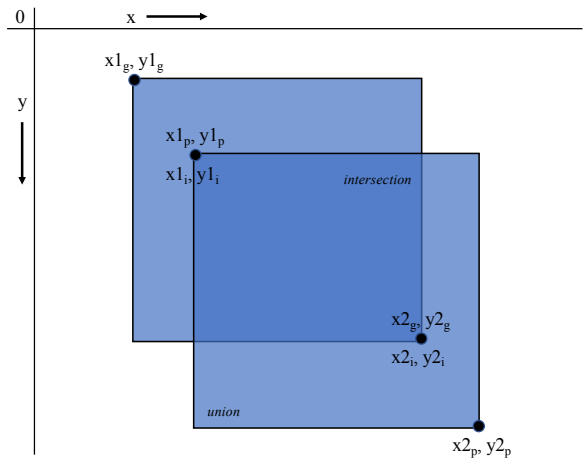

Figure 9. Example illustration of bounding boxes for object detection and their areas of intersection and union.

calculate results accordingly. Having defined IoU, finding the mAP is a function of the precision and recall of the model.

The precision is a measure of the percentage correct predictions out of all predictions. That is,

$$precision = \frac{TP}{TP + FP} \tag{1}$$

where TP, FP, TN, and FN represent true positive, false positive, true negative, and false negative, respectively. The recall is a measure of the number of correct detections found for all present objects. That is,

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

The average precision (AP) is just the area under precision-recall curve. The precision-recall curve is the plot of $p_{interp}(r)$ against the recall r where $p_{interp}(r)$ is the maximum of all precisions $p(\tilde{r})$ at all recall values above r:

$$p_{interp}(r) = \max_{\tilde{r}:\tilde{r} \geq r} p(\tilde{r}) \tag{3}$$

Plotting $p_{interp}(r)$ vs. r instead of $p(r)$ vs. r is shown in Figure 10; using the interpolated precision removes the "zig-zag" nature of the normal graph. This reduces the impact on the AP of small variations in the ranking.

With this background we can now define mAP. In this paper, we use the Area Under Curve (AUC) definition of mAP. The AP is calculated by sampling the curve at all recall values r_{n+1} where the interpolated precision $p_{interp}(r_{n+1})$ is less than $p_{interp}(r_n)$. This means that the average precision is simply the sums of the areas of each "rectangular block" in the curve, as seen in Figure 10. Mathematically,

$$AP = \sum (r_{n+1} - r_n) * p_{interp}(r_{n+1}) \tag{4}$$

Then, the mean average precision is the mean of all APs across all the object classes.

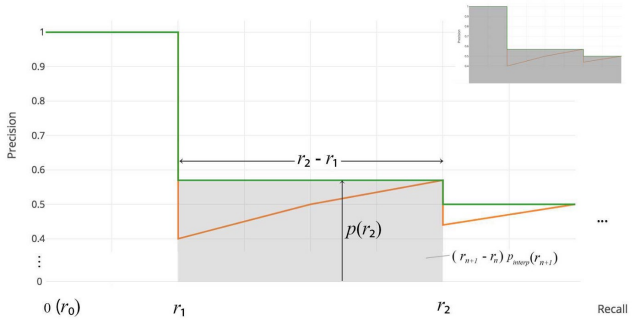

Figure 10. Precision vs. Recall (orange line) and Interpolated Precision vs. Recall (green line) along with shaded area boxes defined by the AUC method for calculating AP.

5.2 SCRAP - Object Detection Results

Here we present the results of comparing the Baseline YOLOv3 model to Model 1 (NRD-retrained YOLOv3) and Model 2 (Transfer-learned YOLOv3). We report the overall mAP score for each of our trained models against the baseline YOLOv3 model as scored on our NRD test set. In addition, we report the precision, recall, IoU, and average precision scores for each model for each waste object class.

Table 1 shows the mean average precision score for the different model types. For Model 1 and 2, we also report the change in mAP with the augmented images training set. Right away we see the poor performance of off-the-shelf YOLOv3 on the NRD dataset. This is expected, since YOLO is not trained to detect waste objects in context; instead, it is trained to detect commonly seen objects like in the ImageNet and COCO datasets. Consequentially, YOLOv3 outputs a an extremely low mAP score of 0.1%. A better benchmark is to look at the mAP for Model 1 (NRD-retrained YOLOv3) which is at 86.75%. This is evidence that the NRD data is capable for the tuning the entire YOLOv3 architecture despite the dataset's small size.

The most significant takeaway from Table 1 is the superior performance of Model 2 (Transfer-learned YOLOv3) against all other trained models. Model 2 is our transfer-learned model as described in Section 4.2.2. The results demonstrate the success of transfer learning in our neural network; allowing the network to retain valuable object-detection information from large and diverse image datasets lends a more precise and accurate detection from the model on the NRD test data.

It is interesting to note that augmentation did not have a significant effect in the mAP scores. This is likely due to the similarity between the training set and the test set in their ambient background, lighting conditions etc. In general, augmentation will be beneficial when the networks are tested in a setting that is different from the training distribution.

Table 1. Mean average precision scores for the baseline YOLOv3 model and all NRD-trained models on the NRD test set, which displays the superior performance of the transfer-learned Model 2.

Model Name	Mean Average Precision (mAP)
Baseline	0.10%
Model 1	86.75%
Model 1 w/ Aug	85.06%
Model 2	**93.77%**
Model 2 w/ Aug	93.24%

Next are Table 2, Table 3, and Table 4 which show the mean precision, recall, IoU, and average precision scores for each model and for each class *cup*, *bottle*, and *paper*, respectively. In these tables overall, Model 2 is again a clear winner in terms of its performance on the NRD test set. For each class, Model 2 has the consistently highest or almost highest precision, recall, IoU, and AP. Hence, Model 2's transfer-learning has allowed it to better detect objects and be more accurate in its predictions.

Generally, in each table, the augmented Model 2 performs second best to Model 2, followed by Model 1 and the augmented Model 1. There are some instances where this pattern doesn't hold, however. For example, the augmented Model 2 has equally good or better recall than Model 2 in the classes *bottle* and *paper*, as seen in Table 3 and Table 4. The success of its recall may indicate that augmentation allows the model to better pick up instances of the waste object that may be out of norm, such as being shaded, hidden, darkened, etc.

Additionally, we see in Table 3 that the baseline YOLOv3 model has a very high precision and IoU for the *bottle* category. We attribute this to the fact that YOLO was trained on a large, diverse dataset; when it is able to detect the bottle, it does a good job of doing so. However, the problem is in its recall—again, since YOLO is not trained for this building-setting, waste-detection context, it fails to recognize most instances of the bottle moving towards the bin.

Figure 11 is a compilation of example outputs of YOLOv3 Baseline, Model 1, and Model 2 on test images from each class. In Figure 11 we see YOLOv3's misclassified or failed detection for each object category. Model 1, as would be expected by the results in Table 1, correctly detects all three objects. Its confidence is high for *bottle* but mediocre for *cup* and *paper*, and has barely-passing IoU scores. Finally we see the outputs of Model 2: a robust detection of each object category and at high confidence. Notably, the confidence and IoU has increased significantly from Model 1 in all three categories.

Figure 11. Visual inspection examples of the detection accuracy in terms of label, confidence, and IoU for each of the recycling items categories. We can see that baseline YOLOv3 works very poorly compared to Model 1 (fully retrained) and Model 2 (Transfer-learned). Model 2 based on Transfer Learning (bottom row) has the best overall performance, making it a suitable object-detection choice for the SCRAP IoWT system.

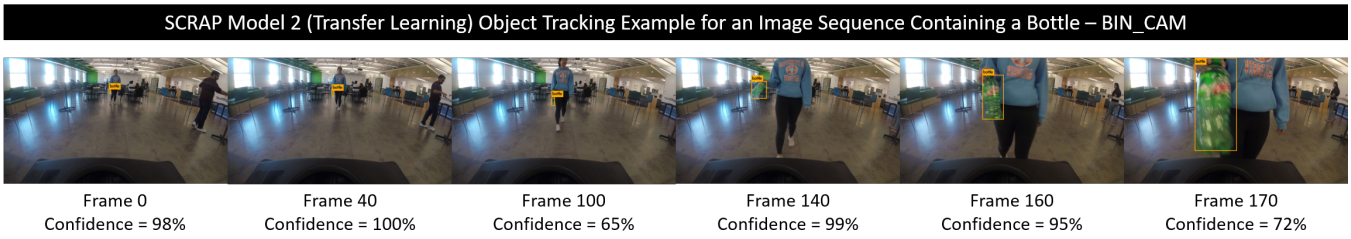

Figure 12. Example tracking output from SCRAP (Model 2). The recyclable item (bottle) is detected well in advance - up to 7-10 feet - as the occupant approaches the location. This gives enough time to activate the visual feedback guidance. In addition, the item is reliably detected throughout the image sequence.

To round out our experimentation results discussion we visualize the real-time object detection, and tracking performance for Model 2 (Transfer-learned) in Figure 12. Figure 12 shows an image sequence across time from the NRD dataset of an individual walking towards a recycling bin with a *bottle* object in hand. We see that even in the first frame, with the individual about 10 feet away from the recycling bin, Model 2 is able to accurately and confidently detect her object. This continues across all frames as the individual walks towards the bin. For each of the 6 frames in this figure, Model 2 detected the object in an average of 137.34 ms. With this speed, we can conclude that real-time waste object-detection will be possible with SCRAP. The system will be able to detect recyclable objects and then assess that detection to prompt the

Table 2. CUP - Mean precision, recall, IoU, and average precision of each model on NRD for the *cup* class. Model 2 (Transfer-learned) has the greatest performance for all metrics.

	Precision	Recall	IoU	AP
Baseline	0.00	0.00	0.00	0.03%
Model 1	0.89	0.77	0.62	83.89%
Model 1 w/ Aug	0.82	0.78	0.58	78.70%
Model 2	**0.93**	**0.93**	**0.65**	**94.35%**
Model 2 w/ Aug	0.91	0.88	0.64	90.76%

Table 3. BOTTLE - Mean precision, recall, IoU, and average precision of each model on NRD for the *bottle* class. YOLOv3 has high precision and IoU but almost zero recall, and the the augmented Model 2 (Transfer-learned) has the greatest performance overall.

	Precision	Recall	IoU	AP
Baseline	**0.96**	0.04	**0.73**	8.24%
Model 1	0.86	0.81	0.59	78.90%
Model 1 w/ Aug	0.80	0.87	0.55	79.74%
Model 2	0.91	0.91	0.65	87.23%
Model 2 w/ Aug	0.90	**0.92**	0.65	**89.96%**

Table 4. PAPER - Mean precision, recall, IoU, and average precision of each model on NRD for the *paper* class. Model 2 (Transfer-learned) has the greatest performance for all metrics.

	Precision	Recall	IoU	AP
Baseline	N/A	N/A	N/A	N/A
Model 1	0.97	0.92	0.73	97.47%
Model 1 w/ Aug	0.96	0.94	0.71	96.80%
Model 2	**0.98**	**0.96**	**0.77**	**99.74%**
Model 2 w/ Aug	0.97	**0.96**	0.73	99.02%

user to dispose of her waste correctly. Ultimately, SCRAP enabled with Model 2 was chosen for a closed-loop deployment test.

For training and testing Model 1 and Model 2 we worked on a machine with the following configuration:

- OS: 64-bit Ubuntu 16.04 LTS
- Processor: Intel Core i7-8700 CPU @ 3.20GHz x 12
- Graphics: GeForce GTX 1070/PCIe/SSE2
- Memory: 15.6 GB
- Disk: 257.9 GB

5.3 IoWT Closed-Loop Deployment

Having successfully trained and evaluated the SCRAP object detection DNN, we next designed and implemented a closed-loop system as the prototype for the Internet of Wasted Things. The system, as depicted in Figure 13, is comprised of a bin-level camera which feeds image sequences to the SCRAP object detector (Model 2) in real-time. The recycling category labels were removed from the waste bins, meaning that when a person approaches, they do not know which bin is reserved for which recycling category (bottles, cups, and paper). As the SCRAP object detector detects the waste item type, an I/O trigger is sent to a LED strip to light up the appropriate bin and provide a visual cue to the person to guide them towards the correct disposal bin for that item. This is implemented through an Arduino Uno R3 used to light up NeoPixel LED strips attached to the bins. As can be seen from Figure 13 top and bottom examples, as the object appears in the field of view of the camera the LEDs on the correct bin light up and persist until the object has been disposed and is out-of-view of the camera. Since the trained SCRAP detector is lightweight and resource efficient, it can be easily run from a NVIDIA Jetson Nano embedded platform connected to a single machine-vision wide field-of-view camera. With the true labels from the recycling bin have been removed, we can exclusively study the effect of the LED visual feedback on the efficacy of the disposal and the effect on minimizing recycling contamination. We are using the test-bed to conduct long-term longitudinal studies and audits for measuring the effectiveness of SCRAP as part of the Internet of Wasted Things.

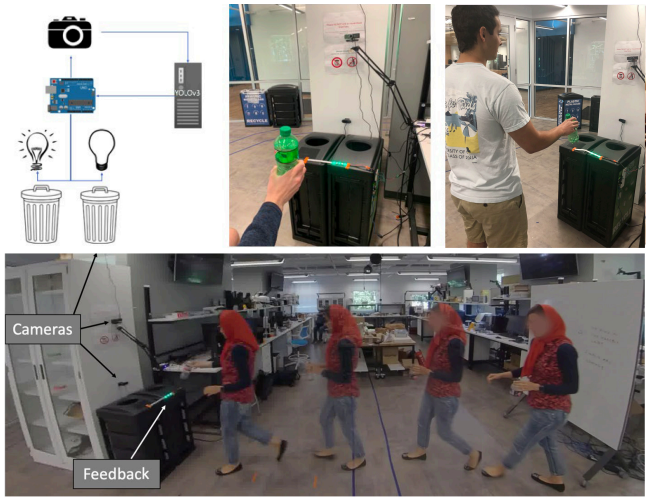

Figure 13. The Internet of Wasted Things Closed-loop Test Bed. As the occupant approaches an unlabeled set of bins, the SCRAP object detector automatically detects the correct object type and provides visual feedback and guidance through LEDs to the occupant for minimizing recycling contamination.

6 Periop Green Dataset - Single-use Surgical Supplies

The Periop Green dataset captures a novel collection of single-use surgical supplies (SUSS) placed on a mock scrub table. The dataset is representative of a real-life surgical setting where this software is expected to be deployed. It serves as a proof-of-concept dataset for training models to validate the computer-vision ability to capture distinct surgical items on a scrub table as they are present before, during, and after a surgery.

6.1 Data Collection

Because the Periop Green application does require real-time detection like the SCRAP project, we had more flexibility in determining camera placement regarding the scrub table. The most important aspect of camera placement was to minimize disruption to the existing surgical team workflow. As one may imagine, the environment of a surgery is very precise; therefore, the Periop Green application would only succeed in the real world if it provides virtually zero disruption to the environment. We decided that a discrete camera placement hovering above the scrub table for the duration of the surgery was the most effective way to achieve this goal. In real life, this may look like a camera being attached to an IV pole that is pointed to the scrub table.

Our mock data was collected in the garage of a doctor leading our team who had access real surgical tools. This was especially helpful during COVID-19 where access to real-life operating rooms was extremely restricted. Items were placed on a table with a blue hospital cover to accurately and adequately mock a surgical scrub table. In actuality, there are upwards of thousands of different item types that could be present on a surgical table. For our proof of concept dataset, we choose to capture and annotate only the following six types:

- Gloves
- Holster
- Knife
- Ligasure
- Stapler
- Suction

As the project expands we seek to incrementally add items to this list. Figure 14 shows what each item type looks like. Each of these items are single-use in a surgery; that is, they are thrown away after surgery if they are opened from their packaging and inside the operating room, despite whether they are actually used.

To increase variety in the training data we took images and videos of each item type individually on the table as well as in aggregate. In general, we tried to capture the following types of data to be representative of what may be seen in an operating room, as well as to test the limits of our model detection:

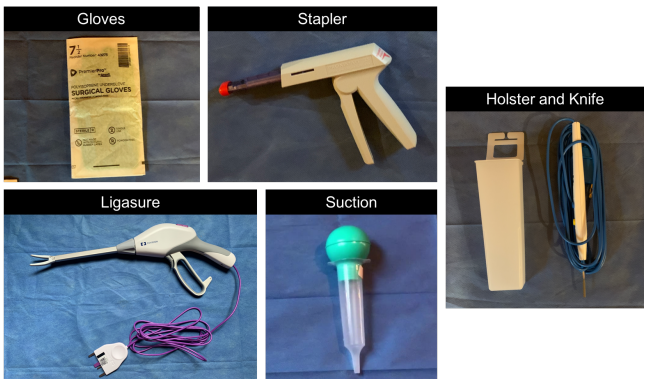

Figure 14. The six item types labeled in the working Periop Green dataset. Each of these items is a single-use surgical tool, meaning they are thrown away after a single surgery. These items can be extremely expensive; the ligasure tool on the bottom left costs $600-$800 each.

1. Images and videos of each item placed individually on the scrub table and in a variety of directions.
2. Images and videos of multiple types of items together on the scrub table, with orderliness and no overlap.
3. Images and videos of multiple types of items together on the scrub table with variety of directional placement and messy overlap.
4. Images and videos of an empty scrub table and with objects not part of our defined set for negative examples.

An example of each data type is shown in Figure 15. In comparison, we show images from an actual operating room in Figure 16. While there are some obvious differences, mostly the lack of the number of items we have available at our disposal and the surrounding environment, our mock data is still representative enough to test proof of model detection ability.

One of the wonderful things about the Periop Green project is the lack of HIPAA concerns regarding data collection. Because our data collection does not seek to capture any faces or sounds, we are not in violation of any medical privacy rules. This will remain true even in real-life hospital settings and bodes well for the future deployment of this project.

6.2 Data Annotation

Annotations for the SUSS dataset were completed on the platform Dataloop [1]. Dataloop was chosen instead of CVAT (used for the SCRAP project) because it better facilitates collaborative annotations through an online interface Like CVAT, Dataloop can be used to draw rectangular bounding boxes, interpolates bounding boxes between frames in a video, and has YOLO text file annotations available for download for images. Figure 17 shows a screenshot from the Dataloop interface.

Figure 15. The four types of data examples included in the Periop Green dataset. Top left: individual items on scrub table; top right: multiple neatly laid items on table; bottom left: disorderly assortment of items on table; bottom right: objects outside item types on scrub table.

6.3 Dataset Organization

The SUSS dataset had a total of 37 videos equalling 7733 images plus 255 pure images. The breakdown of object types in the dataset are shown in Table 5. For training, 64% of the entire dataset was used for training, 16% for validation, and 20% for testing.

Table 5. Breakdown of object types and their relative frequency in the SUSS dataset, as well as where instances of each type were sourced (from video data vs. image data).

	Instances	From video	From images
gloves	8425	97.93%	2.07%
holster	3781	97.46%	2.54%
knife	4093	97.09%	2.91%
ligasure	4131	97.70%	2.30%
stapler	3848	97.56%	2.44%
suction	3672	98.82%	1.17%

7 SUSS Detector

For the Periop Green application we chose to use the YOLOv4 detection model as our baseline instead of YOLOv3. The newest YOLO version is even more state-of-the-art with faster performance and greater detection capability. Learning from the rigorous research done for SCRAP, we have so far only created one trained model on the our SUSS dataset.

Figure 16. Two photos of the scrub table from a real operating room post surgery. There are some obvious differences between real-world photos and our mock dataset, but the layout is similar enough to convincingly establish proof-of-concept.

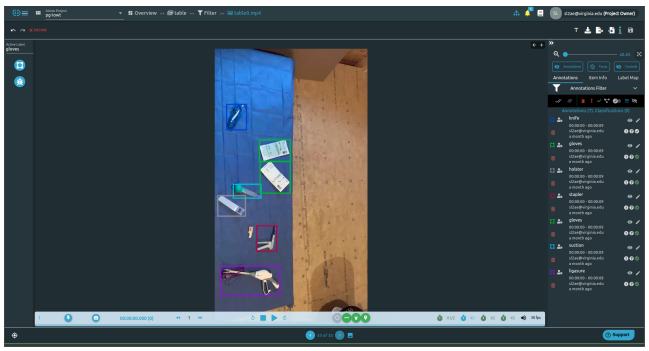

Figure 17. Screenshot of the Dataloop [1] annotation platform used for annotating the SUSS dataset.

7.1 YOLOv4 Object Detector Background

We chose to use the newer version of YOLO, YOLOv4 [5], rather than YOLOv3 as our base model because of its improved performance and speed. YOLOv4 has a much heavier architecture than YOLOv3. Instead of 107 total layers there are 162 layers in the YOLOv4 architecture. The architecture still contains the original darknet layers as seen in Figure 8.

The YOLOv4 layers can also be frozen or re-tuned as indicated in the description of the SCRAP model.

7.2 SUSS Transfer-learned Trained Model

Based on our trial and error results for the SCRAP project, we directly proceeded with creating a transfer-learned model for detection on the SUSS dataset. The reasons for transfer-learning are the same as those described in Section 4.2.2. To recap, transfer-learning allows the "freezing" of lower layers or weights to preserve the feature detection learned from larger image datasets. For our model, we first initialized the training weights with the first 137 layers of the YOLOv4 architecture. Then, we froze the first 105 layers of those before training, and retuned the remaining layers as the model learned on our dataset.

We used the following hyperparameters for training the model:

- Batch = 64
- Subdivisions = 32
- Image width = 416
- Image height = 416
- Channels = 3
- Momentum = 0.949
- Decay = 0.0005
- Learning rate = 0.001
- Max batch iterations = 12,000

Additionally, each convolutional layer preceding each of the three yolo detection layers had a filter of 33, equal to (num_classes + 5) * 3.

For training and testing of the SUSS Detector we worked on the same machine as SCRAP, configuration repeated here for convenience:

- OS: 64-bit Ubuntu 16.04 LTS
- Processor: Intel Core i7-8700 CPU @ 3.20GHz x 12
- Graphics: GeForce GTX 1070/PCIe/SSE2
- Memory: 15.6 GB
- Disk: 257.9 GB

8 SUSS Detector Results

We present the resulting mAP score of our trained model and individual precision and average precision scores for each object type on our test dataset. For the Periop Green application it did not make sense to compare our trained model's results to results from the baseline YOLO weights. This is because our application and object types are so specific to the healthcare industry that no off-the-shelf model comes trained to identify such types of objects. However, this means that our object detector created is a novel computer vision application with potentially far-reaching impact. The metrics shown are the same as defined in Section 5.1.

Results for our model are shown in Table 6 and Table 7. Detection time on the test set took 42 seconds.

Table 6. The mean average precision for our SUSS Detector given an IoU threshold of 50%.

	mAP
SUSS Detector	**96.61%**

Table 7. The true positive, false positive, precision, and average precision for each object type as measured on the test set of the SUSS data using the SUSS Detector.

	TP	**FP**	**Precision**	**AP**
gloves	1565	114	93.21%	94.87%
holster	716	60	92.27%	95.09%
knife	760	69	91.68%	96.87%
ligasure	804	124	86.64%	97.54%
stapler	742	36	**95.37%**	97.62%
suction	711	42	94.42%	**97.69%**

8.1 SUSS Detection Examples

We tested the SUSS Detector on many examples to evaluate its performance. Figure 18 shows some images of the detector in action. It is clear that for images similar to the dataset, our model performs extremely well. The confidence scores for detections are at or near 100% for most of the object types. Figure 19 shows the model in action on an image from the real operating room, where our model was able to detect a suction object on the table. However, generally, our model did not perform well on real-life OR photos, likely because of the environment differences as discussed previously.

However, there are some interesting patterns we saw with our model detection. The model performed very poorly on images outside of our dataset, even if they were seemingly similar in appearance. This was especially confusing because these were simple images rather than the ones with messy overlaps as shown in Figure 18. We determined that the cause of this was because most of our dataset was sourced from videos rather than images. Looking at Table 5, we see that over 97% of all object type instances come from video frames as images rather than pure images. This means that the model learned to detect objects with a high amount of blur present rather than on a sharp image. To test our theory we ran the model on an original image and then applied artificial blur to the image and ran the model again. The results are seen in Figure 20, where it is clear that our detector performs much better on a blurry photo.

8.2 Tracking Used vs. Unused Supplies

The ideal goal of the Periop Green project is to track single-use sterile surgical supplies (SUSSS). That is, we'd like to detect which items on the table have been used or not used during the surgery. This way, we can provide targeted recommendations for reducing waste and saving money in operating rooms. Detecting item use with our trained YOLO

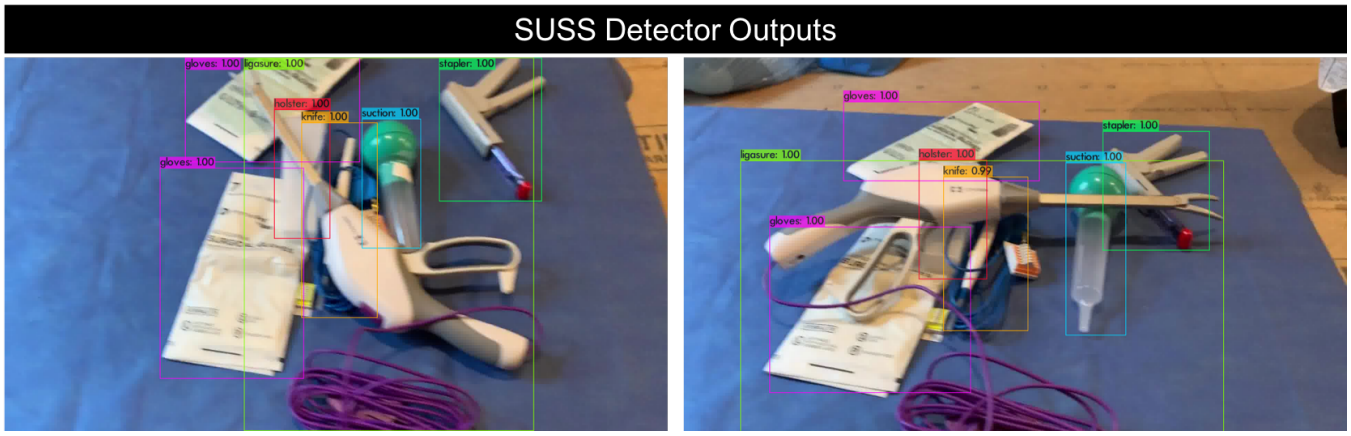

Figure 18. Outputs on the SUSS detector model on images from the SUSS dataset test set. The model performs extraordinarily well on these images, capturing every object present with high confidence.

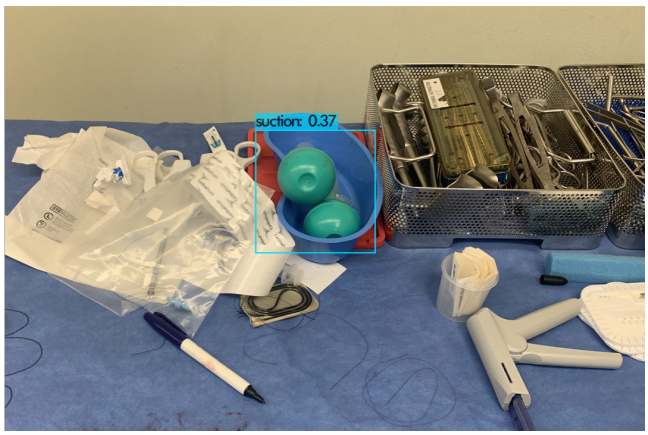

Figure 19. The output of our detector on an image taken in a real-life operating rooom after a surgery. The model correctly identifies the suction on the table with some confidence but notably misses the stapler.

model outputs is not a trivial endeavor. Our approach, given a video to analyze, is as follows:

1. Graph the number of instances of each object type at each frame.
2. Detect, based on some threshold, when the number of instances of an object has decreased for an interval long enough to be convinced that the object has left the table.
3. If an object has left the table during a surgery, then it is considered to have been used during the surgery.

We are still in the exploratory phases of this approach and more rigorous analyses must be conducted to determine whether this is the appropriate path to take, how well it works on our dataset, etc. An example graph is shown in Figure 21 for what the above described graph may look like.

9 Future Work

Both of the described applications have many avenues for future direction. The possibilities only increase with greater use of IoT connectivity and machine learning integrations.

SCRAP

In the future we envision the following areas of work:

- Field deployment and controlled testing.
- Embedded computing: expanding and refining the standalone working component.
- Preserving privacy in processing.
- Deploy federated learning across many locations.
- Detection contamination of the materials.
- Detection intention of the individual.

Periop Green

For Periop Green, we'd like to tackle the following problems in the future:

- Refining the detection of used vs. unused objects.
- Comparing items on the scrub table with a pre-made pdf list of items present before all surgeries.
- Prioritizing items of high-yield, i.e. ones that are very expensive.
- Augmenting the data set through typical augmentation or 3-D modeling.
- Expanding the dataset of items.

10 Conclusion

The paper presents the foundations of building the Internet of Wasted Things (IoWT). We have explored two potential avenues for application of the IoWT: namely SCRAP and Periop Green.

For SCRAP, we begin with first collecting and annotating a new Naturalistic Recycling Dataset (NRD) which addresses the limitations of other trash-related datasets by capturing

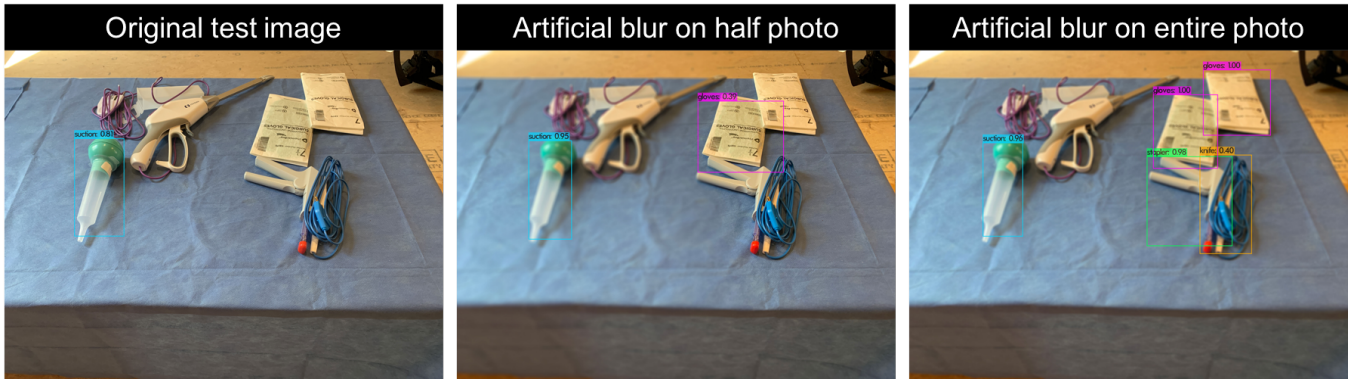

Figure 20. Performance of our model on a single image with and without artificial blur applied. The detector performs extremely poorly on the original image, which is quite sharp, but performs markedly better on the blurry photo. This is likely because the model's data was sourced from videos converted to images, i.e. blurry photos.

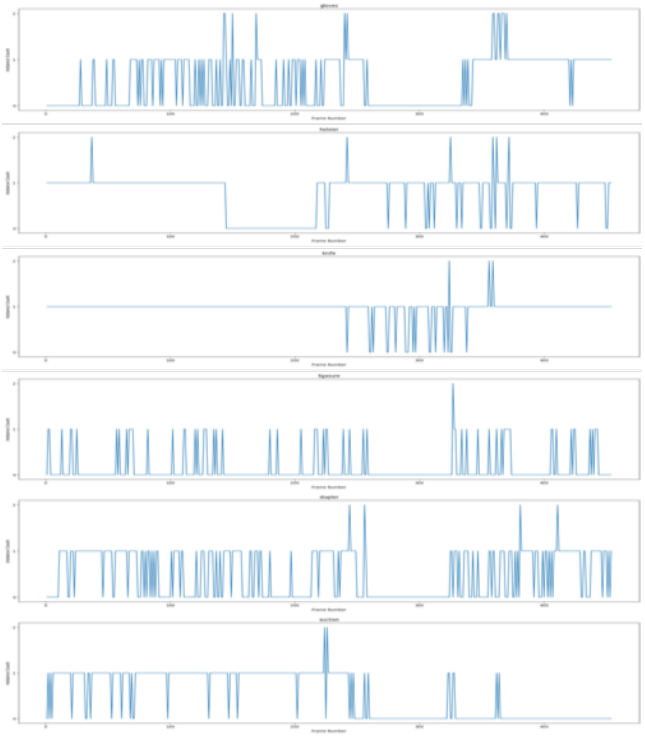

Figure 21. Example output graphs for the SUSS Detector on each object type to identify used vs. unused items. The graph shows the number of instances of the object as identified by the model vs. the frame number of the video. In theory, used items would see a dip in their number of instances over an extended period of frames, indicating that the item had left the table and been used.

the naturalistic occupant behavior as they approach a recycling bin location inside a building. Next, we designed and implemented our automated waste item detection and guidance system. SCRAP object detection is trained on the NRD using transfer-learning on top of the Darknet-53 feature detector and YOLOv3 layered architecture. We demonstrate the efficacy and performance of SCRAP as compared to two other object detection methods to find that SCRAP (Model 2) performs the best in terms of its mean average precision of waste object detection, up to 93%. Finally we also design and implement a closed-loop visual feedback system that can be used for deploying SCRAP in a low-cost and scalable manner without modifying any of the existing bins.

Secondly, with Periop Green we also collected and annotated our own custom single-use surgical supplies (SUSS) dataset, and used that dataset to create a SUSS detector. Both of these contributions are novel research as this project approach has never been attempted before. Our detector has a mean average precision of over 96% on our test data and shows great promise for detection in real-world operating rooms.

By combining smart recycling and AI-powered waste management, organizations can streamline the entire process from garbage collection to disposal. This has positive effects on sustainability by reducing the volume of incorrectly disposed objects and also minimizing recycling contamination. We believe our research is a remarkable and instrumental step in the creation of a IoWT system that can be expanded in a variety of interesting and impactful avenues.

References

[1] 2020. Dataloop AI. https://dataloop.ai/
[2] Russell Allan et al. 2018. China puts recycled to the sword. *Appita Journal: Journal of the Technical Association of the Australian and New Zealand Pulp and Paper Industry* 71, 3 (2018), 202.
[3] Andreas Bartl. 2014. Moving from recycling to waste prevention: A review of barriers and enables. *Waste Management & Research* 32, 9_suppl (2014), 3–18.
[4] Yoshua Bengio. 2012. Deep learning of representations for unsupervised and transfer learning. In *Proceedings of ICML workshop on unsupervised and transfer learning*. 17–36.

[5] Alexey Bochkovskiy, Chien-Yao Wang, and Hong-Yuan Mark Liao. 2020. YOLOv4: Optimal Speed and Accuracy of Object Detection. arXiv:2004.10934 [cs.CV]
[6] G. Bradski. 2000. The OpenCV Library. *Dr. Dobb's Journal of Software Tools* (2000).
[7] Amy L Brooks, Shunli Wang, and Jenna R Jambeck. 2018. The Chinese import ban and its impact on global plastic waste trade. *Science advances* 4, 6 (2018), eaat0131.
[8] Charlotte Collins. 2019. The Global Environmental Recycling Crisis. (2019).
[9] Berardina De Carolis, Francesco Ladogana, and Nicola Macchiarulo. 2020. YOLO TrashNet: Garbage Detection in Video Streams. In *2020 IEEE Conference on Evolving and Adaptive Intelligent Systems (EAIS)*. IEEE, 1–7.
[10] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. 2009. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*. Ieee, 248–255.
[11] Jonathan Hui. 2019. mAP (mean Average Precision) for Object Detection. https://medium.com/@jonathan_hui/map-mean-average-precision-for-object-detection-45c121a31173
[12] Alexander B. Jung, Kentaro Wada, Jon Crall, Satoshi Tanaka, Jake Graving, Christoph Reinders, Sarthak Yadav, Joy Banerjee, Gábor Vecsei, Adam Kraft, Zheng Rui, Jirka Borovec, Christian Vallentin, Semen Zhydenko, Kilian Pfeiffer, Ben Cook, Ismael Fernández, François-Michel De Rainville, Chi-Hung Weng, Abner Ayala-Acevedo, Raphael Meudec, Matias Laporte, et al. 2020. imgaug. https://github.com/aleju/imgaug. Online; accessed 01-Feb-2020.
[13] C Lieber. [n.d.]. Hundreds of US cities are killing or scaling back their recycling programs. https://www.vox.com/the-goods/2019/3/18/18271470/us-cities-stop-recycling-china-ban-on-recycles.
[14] Tsung-Yi Lin, Priya Goyal, Ross Girshick, Kaiming He, and Piotr Dollár. 2018. Focal Loss for Dense Object Detection. arXiv:1708.02002 [cs.CV]
[15] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. 2014. Microsoft coco: Common objects in context. In *European conference on computer vision*. Springer, 740–755.
[16] J Margolis. [n.d.]. Mountains of U.S. recycling pile up as China restricts imports. https://www.usatoday.com/story/news/world/2018/01/02/mountains-u-s-recycling-pile-up-china-restricts-imports/995134001/
[17] NVIDIA, Péter Vingelmann, and Frank H.P. Fitzek. 2020. CUDA, release: 10.2.89. https://developer.nvidia.com/cuda-toolkit
[18] Ryan T O'Connor, Dorothea C Lerman, Jennifer N Fritz, and Henry B Hodde. 2010. Effects of number and location of bins on plastic recycling at a university. *Journal of applied behavior analysis* 43, 4 (2010), 711–715.
[19] Pedro F Proença and Pedro Simões. 2020. TACO: Trash Annotations in Context for Litter Detection. *arXiv preprint arXiv:2003.06975* (2020).
[20] Joseph Redmon. 2013–2016. Darknet: Open Source Neural Networks in C. http://pjreddie.com/darknet/.
[21] Joseph Redmon, Santosh Divvala, Ross Girshick, and Ali Farhadi. 2016. You Only Look Once: Unified, Real-Time Object Detection. arXiv:1506.02640 [cs.CV]
[22] Joseph Redmon and Ali Farhadi. 2016. YOLO9000: Better, Faster, Stronger. arXiv:1612.08242 [cs.CV]
[23] Joseph Redmon and Ali Farhadi. 2018. YOLOv3: An Incremental Improvement. *arXiv* (2018).
[24] George E Sakr, Maria Mokbel, Ahmad Darwich, Mia Nasr Khneisser, and Ali Hadi. 2016. Comparing deep learning and support vector machines for autonomous waste sorting. In *2016 IEEE International Multidisciplinary Conference on Engineering Technology (IMCET)*. IEEE, 207–212.
[25] William H Shrank, Teresa L Rogstad, and Natasha Parekh. 2019. Waste in the US health care system: estimated costs and potential for savings. *Jama* 322, 15 (2019), 1501–1509.
[26] Bulk Handling Systems. 2019. Max-AI. https://www.max-ai.com/
[27] Gary Thung and Mindy Yang. 2017. Classification of Trash for Recyclability Status. https://github.com/garythung/trashnet
[28] ZenRobotics. [n.d.]. Robotic Waste Recycling Solutions. https://zenrobotics.com/