A PORTABLE SYSTEM TO SORT, DETECT, AND CLASSIFY MICROPLASTICS

by

WARREN SMITH

MARK CHENG, COMMITTEE CHAIR
AIJUN SONG
QIANG HUANG

A THESIS

Submitted in partial fulfillment of the requirements for the degree of Master of Science in the Department of Electrical and Computer Engineering in the Graduate School of The University of Alabama

TUSCALOOSA, ALABAMA

2023

ABSTRACT

The mass pollution of plastics in the world is having a detrimental impact on both marine biology and human health. While there are efforts to clean and analyze plastics with the help of the community, these plastics often break down into smaller particles referred as microplastics, which are more difficult to study and remove. Numerous studies are coming out with the attempt to detect and identify these microplastics with hopes for future improvements to reduce this global hazard. However, a main issue with current research is the lack of standardization in the testing and sampling of microplastics. Most studies involve collecting samples from different bodies of water and then transporting them to a laboratory for treatment, filtering, and examination using various methods to classify the type of plastic and gather information about its size and quantity. This process can take a large amount of time and is separated primarily into stages: sampling, treatment, classification, and detection.

This thesis will explore different research techniques that incorporate some of these stages and proposes a portable system that can sort particles using mechanical filtration, detect particles using the You Only Look Once (YOLO) convolutional neural network, and classify particles using Raman spectroscopy. The system will be deployed in the field with a focus in reducing the amount of time and labor involved with the process of microplastic analysis by combining techniques into one system. The project that will be presented in this thesis is a work in progress and research is still being implemented and advanced within the current system. Possible improvements are addressed for future students to expand on. The goal of

this work is to lay a foundation for a sustainable, versatile system for use in various environments and over extended periods of time. This thesis will present a literature review on the current work being done in the field as well as the techniques used, and each individual part of the current deployed system will be explained.

DEDICATION

This thesis is dedicated to my parents and my future wife.

For their endless love, support, and encouragement.

LIST OF ABBREVIATIONS AND SYMBOLS

PET	Polyethylene terephthalate
HDPE	High-density polyethylene
PVC	Polyvinyl choride
LDPE	Low-density polytheylene
PP	Polypropylene
PS	Polystyrene
PA	Polyamide
YOLO	You only look once
CNN	Convolutional neural network
RCNN	Region-based convolutional neural network
IoU	Intersection over Union
PCI	Peripheral component interconnect
COM	Component object model
AVR	Alf and Vegard's RISC
μm	Micrometer
mm	Millimeter
TB	Terabyte
KB	Kilobyte
GUI	Graphical user interface
PLA	Polyactic acid

UV	Ultraviolet
V	Volts
DC	Direct current
A	Amps
mA	Mili amps
AI	Artificial intelligence
fps	Frames per secon

ACKNOWLEDGMENTS

Any work done in this project would not be possible without the expertise of my supervisor Dr. Mark Cheng. I give all my thanks to him for giving me this opportunity and giving me his support, supervision, and guidance throughout. He is a great professor to look up to.

I am grateful for the current and previous students I have worked with in the lab: Wyatt Brusch, Fairoz Abida, Drew Burjek, Pratiksha Chaudhari, Rong Fu, Tauhidul Haque, Savannah Wooldridge, Aiden Leyen, and Eric Smallwood. Their support and friendship throughout graduate school has helped me beyond measure.

I would like to thank Yongli Wager and Yongtao Yao for their guidance and support in our weekly meetings.

I would like to thank the employees at The Cube and at Gorgas library for printing out the 3D designs I have submitted throughout the research.

CONTENTS

LIST OF TABLES

LIST OF FIGURES

CHAPTER 1 LITERATURE REVIEW

1.1 Introduction

Microplastics are a pervasive and persistent form of pollution that has been found in oceans, rivers, soil, and in the food and water that we consume. They are small plastic particles that measure less than 5 millimeters in size. Microplastics come from the breakdown of larger plastic objects by physical, chemical, and biological factors which are considered as secondary microplastics, as well as the shedding of synthetic clothing fibers, and the use of personal care products that contain microbeads which are considered as primary microplastics [1,2]. From 1950 to 2018, the global production of plastics increased from 1.3 million tons to 359 million tons [3]. Plastic pollution has been found in every corner of the world's oceans, with estimates suggesting that there are more than 5 trillion pieces of plastic debris (totaling over 250,000 tons) floating in the seas [4]. There is a wide range of shapes within microplastics, including pellets, irregular shapes (fragments), fibers, film, and foam. These shapes have different mass-surface area ratios and are difficult to pinpoint without further improvement on extraction methodologies [5].

The impact of microplastics on the environment and human health is not fully understood, but there is growing concern about their potential effects such as the introduction of toxins, increasing the transmission of diseases, blocking sunlight from photosynthesizing marine organisms, and encouraging the development of hypoxic areas in the ocean [6,7]. Microplastics have been shown to absorb toxins and transport them

throughout the food chain, potentially affecting the health of marine life and humans who consume seafood [8, 9]. In addition, microplastics have been found in drinking water raising concerns about their potential impact on human health [4, 10]. They have been found in numerous consumer products such as beer, honey, sugar, and salt [11, 12, 13, 14]. Multiple occurrences of microplastics have been discovered in the stomach, soft tissue, and gastrointestinal tract of shellfish and fish as well which can be commercial interested food items [4]. Research into human consumption is being done with there being evidence of microplastics in human placentas as well as changes to chromosomes which result in affects to health such as infertility, obesity, and cancer [15, 16].

One of the major challenges being solved in microplastic analysis is uncovering their source and distribution. There is a demand in learning more about the concentration and type of plastic that is in our environment. Current methods for tracking and measuring microplastics are labor-intensive and time-consuming, making it difficult to accurately assess their impact and develop effective management strategies. The construction of a system that can efficiently filter and analyze microplastics at a high throughput will be beneficial for the future of microplastic research. To address the issue of microplastics, there is a need for further research to better understand their sources and distribution, as well as the potential impacts on the environment and human health. This research could include studies on the effectiveness of current waste management practices in reducing the release of microplastics into the environment, as well as the development of more effective methods for tracking and measuring microplastics such as the research in this thesis. Analysis of microplastics is essential for understanding their potential impacts on the environment and human health. Common sampling practices that are used in the analysis of

microplastics will be presented along with research backing the implementations that are used in this research regarding classification through chemical identification and object detection.

1.2 Sampling Methods

The primary representative of the results in each study into microplastics is the sampling location. Microplastics are found in oceans, freshwater, soil, air, and the food/water that we consume [17]. Microplastics can affect soil aggregation as well as the flux of greenhouse gases when they are accumulated in the soil [18]. This research tests primarily freshwater bodies currently and mismanaged waste like the littering from municipal solid waste is considered as the primary source of microplastics in freshwater ecosystems [19]. An estimated average in Lake Superior, Huron, and Erie was 43,000 particles km^{-2} sampled with a neuston net [20] as well as a reporting that surface waters of Lake Michigan had an average abundance of 17,000 particles km^{-2} which was sampled with a mantra trawl [21]. Urban rivers such as the Los Angeles River and the San Gabriel River in Southern California have been sampled using a mantra trawl, different sized hand nets for the shore, and a streambed sampler that showed findings of 12,652 microplastics between 1 and 4.75mm [22]. Microplastics can enter these environments from direct release of microplastics through washing and personal care, the breakdown of larger plastics from waste, discharge from industrial processes, sewage treatment plants, stormwater runoff, agricultural practices, tire wear, and airborne transport [17]. Simulation techniques are being researched into that can help understand the affects, transport, and occurrence of microplastics throughout the world [23, 24]. More research into varied bodies of water will improve the analysis in microplastics.

Microplastics can be sampled using a variety of methods, including water column, sediment, or biofouling sampling [25]. Water column sampling involves collecting water from the surface or at different depths in the water column, filtering it, and analyzing the filtered sample for the presence of microplastics. Sediment sampling involves collecting sediment from the bottom of a water body and analyzing it for microplastics. Biofouling sampling or bioaccumulation involves collecting samples of organisms that have accumulated microplastics within their tissues.

Water column sampling is the focus in this research and there are multiple ways to separate the microplastics from the water such as using mechanical, chemical, or biological filters. Mechanical filters use a physical barrier to remove microplastics from water. These filters can be a mesh screen or complex system with multiple layers of filtration to capture microplastics of different sizes such as the strainers used in this research to obtain microplastics within the range (100-300 µm). Examples of mechanical filters incorporated are the neuston nets [26] and manta trawls [27] used for floating microplastics on sea surfaces. Chemical separation uses chemicals to bind to and remove microplastics from the water such as cationic surfactant-modified zeolite that is shown to remove a large amount of PE and PA [28]. This process uses compounds that are attracted to the surface of the microplastics from water, such as activated carbon or alumina. A chemical like polyacrylamide can be used to extract microplastics by creating a gel like formation that traps microplastics. Biological separation uses microorganisms to break down and remove microplastics from water which are primarily used for wastewater [29]. This process typically relies on bacteria or fungi that digest and break down microplastics into biomass or other forms of degradation.

One consideration in the analysis of microplastics is sample preparation. It is essential to ensure that the sample is adequately prepared to have accurate results. This typically involves homogenizing the sample to ensure that it is evenly distributed, and then using a filtering technique to remove any larger particles that may interfere with the analysis. Once the sample is prepared, it can then be analyzed further and classified into its morphology or type of plastic [25].

1.2.1 Sorting

Incorporating microfluidic chips that apply forces to sort micro particles are gaining a lot interest in the field of microplastic analysis. There are passive and active methods used for microfluidic designed sorting of microplastics. Passive methods use structures in the flow channel that force microparticles into their distinct output such as hydrodynamic filtration, electrostatic sieving, bacteria chemotaxis, deterministic lateral displacement, straight inertial microfluidics, and spiral microfluidics [30]. Active methods use the exertion of applied forces on microparticles such as optofluidics, field flow fractionation, electrophoretic, dielectrophoretic microfluidics, and acoustics [30, 31, 32]. Dielectrophoresis (DEP) is becoming one of the most popular techniques as it can easily sort different sizes with the most efficient size-based isolation of microparticles. This technique uses electrostatic forces to control and navigate micro particles.

Density sorting mechanisms are useful when dealing with untreated water that is filled with debris such as sediment or sand because of the weight difference compared with microplastics. A 3D printed hydro cyclone that sorts microplastics from sediment based on density is a key element within the filtering mechanism in this work. This device uses centripetal forces generated by the use of circular flowing water within a cylinder-conical unit. A

representation of a hydro cyclone can be seen in Figure 1.1 which displays the circular pathing of particles through the unit.

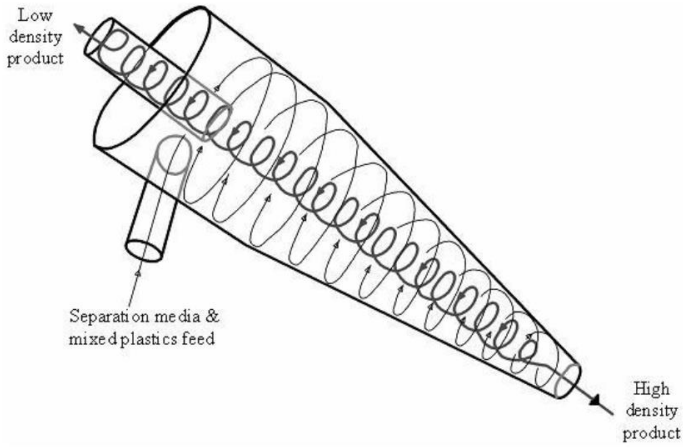


Figure 1.1 Hydro Cyclone [33]

This fluid sample enters horizontally through the entry inlet and exits through the two outlets at the top and bottom of the cyclone. Used with a high flow rate, the entering water produces a vortex within the device which cause denser particles to circulate at the edge of the unit that causes fiction and reduces the kinetic energy of the particles. This allows for denser material to flow out and downward through the bottom outlet, while lighter material flows inward and upward through the top outlet. The upward and downward flows are referred to as overflow and underflow respectively. This device can have benefits such as a modular design, low weight, simple operation, and optimal separation characteristics [34].

1.3 Visual Identification

One common method of microplastic analysis is visual identification using microscopic examination, which involves using a microscope to identify and count the number of microplastics present in a sample. This is helpful to examine the physical characteristics of microplastics, such as their shape, color, and texture. Visual identification can include light

microscopes, electron microscopes, or scanning probe microscopes [35]. This method is useful for determining the overall concentration and size of microplastics in each sample but does not provide information about the type of plastic present. This technique will be used in this research to count, label, and collect the size of all objects that pass through the system as well as to initiate the Raman scanning.

Most research in the visual identification of microplastics is moving forward with computer vision implementations such as object detection for collecting the count, morphology type, and size of micro plastics. Object detection is a computer technology related to computer vision and image processing that deals with detecting instances of objects in images [36]. Object detection algorithms typically use machine learning techniques to learn the characteristics of objects from a training dataset. The trained model is then applied to a new image or video to identify and locate objects in the scene, which is referred to as inference. There are many different algorithms and approaches to detecting objects in an image, but they generally have the tasks of classification, localization, and segmentation. Classification algorithms identify the class of an object (e.g., plastic, dirt, algae), while localization algorithms provide the coordinates of the object in the image (e.g., the bounding box of the object in an image). Segmentation is used to locate commonality of pixels in certain areas of the frame to distinguish curves, lines, and objects [36]. Another type of detection is known as instance segmentation where instead of a bounding box around the detected object, an outline of the object is made which gives a much better representation of the size for the object in question. Examples of these methods are shown in Figure 1.2.

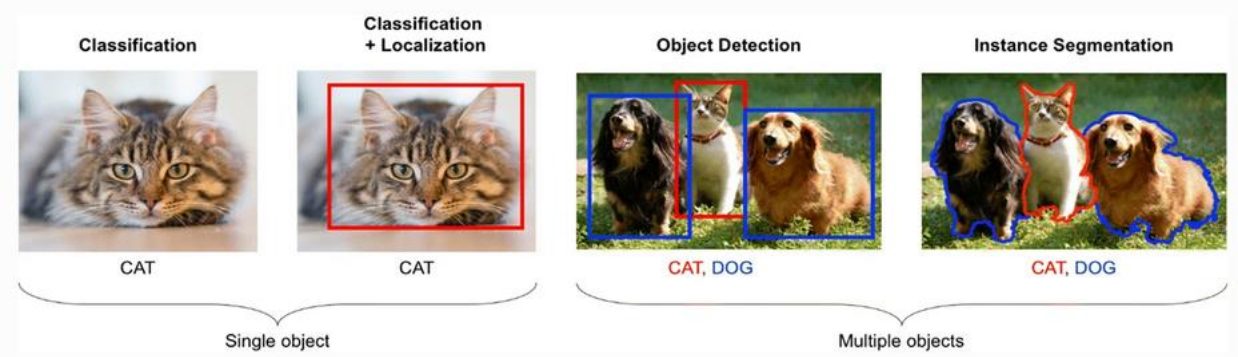


Figure 1.2 Computer Vision Methods [37]

More recent computer vision algorithms primarily use convolutional neural networks (CNNs) to improve performance and accuracy. These algorithms can learn complex features and patterns in the training data and can process images much more efficiently than the basic sliding window approach. CNNs primarily use a combination of convolutions, pooling, and activation functions such as the Rectified Linear Unit (ReLU) to convert input data into an output tensor. The filter size primarily details the spatial extent in the convolution action which extracts features of the image and update weights by convolving the filter across the data. Parameters such as stride, depth, and zero-padding dictate the size of the output of the convolutions [36]. The convolution operation can be seen in the equation below where a dot product is performed between weights (w) and the spatial input (x) with an added bias term (b) set in a non-linear activation function (f). The filter size is represented as (d), (y^l_i) represents the output of the i^{th} neuron in the layer l, and d1/d2 are the filter widths and heights.

$$y_{ij}^{l} = f\left(b_{ij} + \sum_{k=0}^{d1-1}\sum_{l=0}^{d2-1} w_{(i+k)(j+l)} \times x_{(i+k)(j+l)}\right)$$

The activation function that is applied within the convolution primarily is the ReLU which has the primary function of max(0,x). The output then goes through a pooling application

which is a down sampling along the spatial dimensions which helps extract prominent features in the image. The primary pooling method applied is max pooling. Depending on the application, either average-pooling or sum-pooling is chosen. Different neural network architectures will use a combination of these techniques in a cascading formation with different parameters of design for the needed application. A generic example of this architecture is displayed in Figure 1.3.

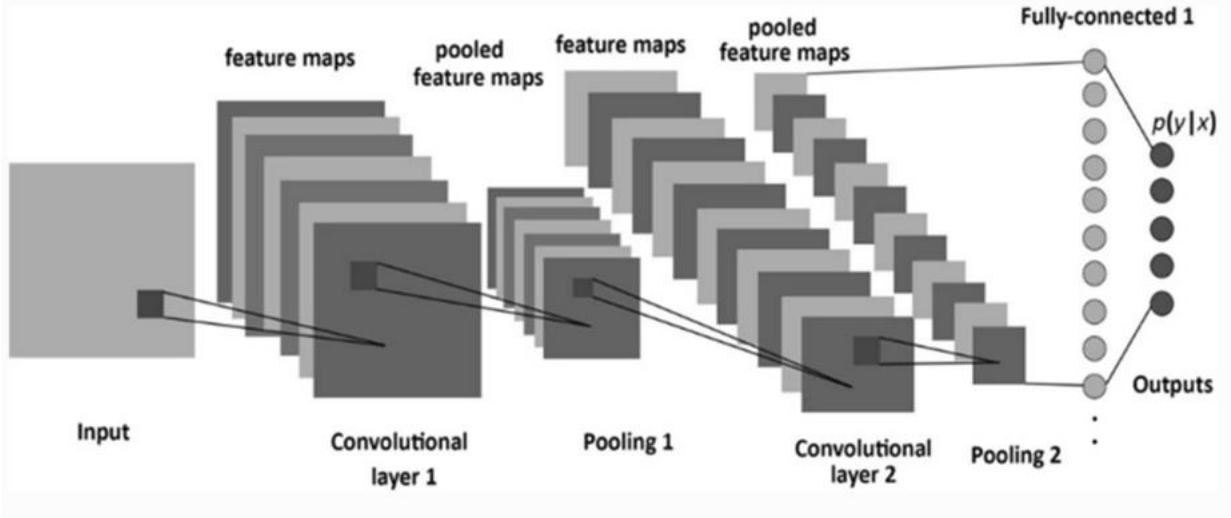


Figure 1.3 Neural Network Architecture [37]

The two types of problems object detection tackles are primarily regression and classification issues. Two stage object detectors like the R-CNN and Fast R-CNN firstly generates the region of interest using a region proposal network while the second stage predicts the objects and boxes for the regions. Single stage object detectors like the SSD, RetinaNet, and M2Det have a simpler architecture and are designed to output the bounding boxes and class probabilities by considering all region proposals or spatial sizes of an image. Deep learning can provide more efficient results when using highly featured objects and multiple categories but does have limitations as well such as multi-scale training, foreground-background class

imbalance, detection of small objects, necessity of large datasets and computational power, and inaccurate localization during predictions [37].

1.3.1 YOLO

What separates the YOLO model from other choices is its combination of speed, accuracy, and ease of use. It is a single stage object detector that gives the improvement of speed over accuracy compared to the two stage detectors [38]. The YOLO algorithm works by dividing the input image into a $C \times C$ grid of cells, and each cell is responsible for predicting a set of bounding boxes that can contain objects. Each bounding box is associated with a probability that represents the confidence that an object is present in the box as well as the position and dimensions of the box. To make predictions, the YOLO algorithm uses a multi-scale detection approach, where it looks at the image at different scales and at different locations in the image. This allows it to detect objects of different sizes and in different positions in the image. Each grid predicts an array set of parameters that can be represented in Figure 1.4 which hold the initial five bounding box parameters such as the probability an object is in the grid (p_c), the center of the bounding box (b_x and b_y pixel coordinate), and the dimension of the bounding box (b_h height and b_w width). The class probability that an object belongs to the i^{th} class fills the rest of the array for the grid. Each grid cell predicts $(B \times 5 + n)$ values, where n is the number of classes and B is the number of bounding boxes. The output tensor shape is then $C \times C \times (B \times 5 + n)$. The representation displays an example image frame divided into 19x19 grids with an array representation of one of the grid cells that show an example of 4 bounding boxes within the cell.

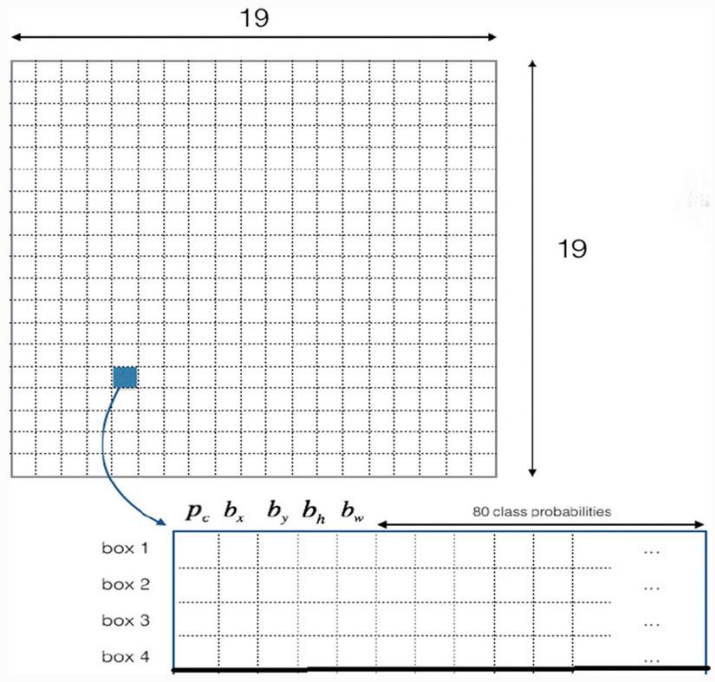


Figure 1.4 Grid Cell Array Representation [37]

A confidence score is computed for each bounding box by multiplying the probability an object is in the grid (p_c) with Intersection over Union (IoU) between the ground-truth and predicted-bounding-box. A class score is then computed to represent the probability of a certain class appearing in the box and the accuracy of the predicted box fitting the object. The total bounding box sizes are determined based on the anchor boxes set for the objects that are being trained on. There is a limit size with anchor boxes such as the smallest 32×32 anchor box the RetinaNet can use. This is information needed for detecting objects that are outside the constraints of the objects within the training library. One would need to add images of the objects differing in size or edit the objects total size within the image for the training library. This preset anchor box height and width from the objects annotated in the library dictates the bounding box formation.

Grid cells that are adjacent to each other can get a prediction on the same object with differing center locations which will give overlapping bounding boxes. These overlapping bounding boxes will need to be removed to display an efficient detection. Most bounding box predictions will be removed based on a confidence threshold set with the program such as any bounding box confidence being lower than 50%. Non max suppression is also used based on the IoU to remove any left-over overlapping boxes. The overlapping boxes that have an IoU greater than a set threshold will be repeatably removed until there are no other boxes that have a lower confidence than the bounding box with the maximum score that is initially selected to compare. A representation of non max suppression with an illustration of the IoU method is displayed in Figure 1.5.

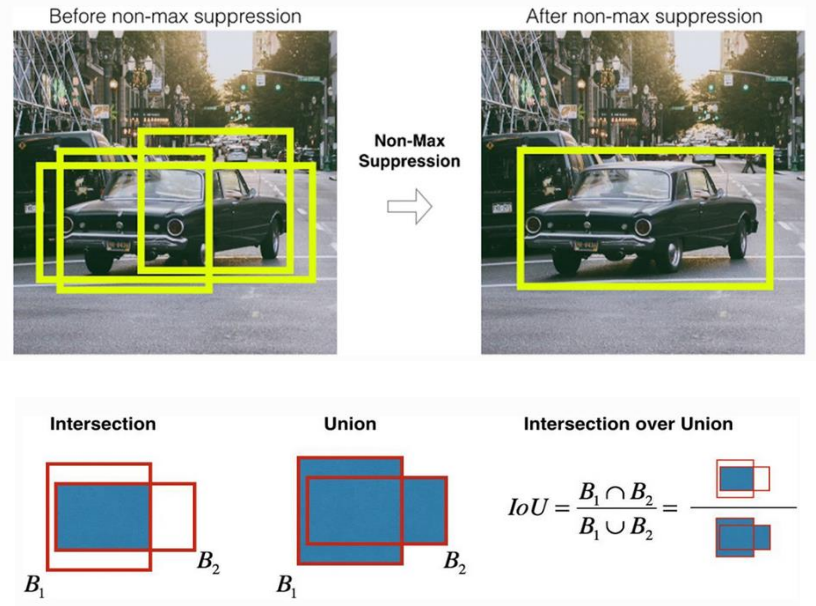


Figure 1.5 Non-Max Suppression & IoU Representation [37]

The YOLO architecture initially was inspired by the GoogLeNet architecture which uses multiple filters with two convolutional layers, two max-pooling layers, and inception modules. In the YOLO version, the inception modules are replaced with (1× 1) convolutions and then (3 × 3)

convolutional filters and the first convolutional layer that has a (7 × 7) filter which is also referred as the head of the neural network. There are twenty-four convolution layers preceded by two fully connected layers. Four of these convolutional layers are followed by max-pooling layers [37]. There is a (1 × 1) convolution and average pooling throughout. Leaky Rectified Linear Unit (LReLU) is used for each layer except the last which uses a linear activation function. The overall structure for the feature extraction network used is considered as the backbone of neural networks in which YOLO is primarily based on GoogLeNet. The loss function is used to fit the model to the given training data by comparing the target and predicted output values. YOLO uses a mean squared error (MSE) which finds the average of the squared difference between the predicted output and the target. An example of the original YOLO architecture can be seen in Figure 1.6.

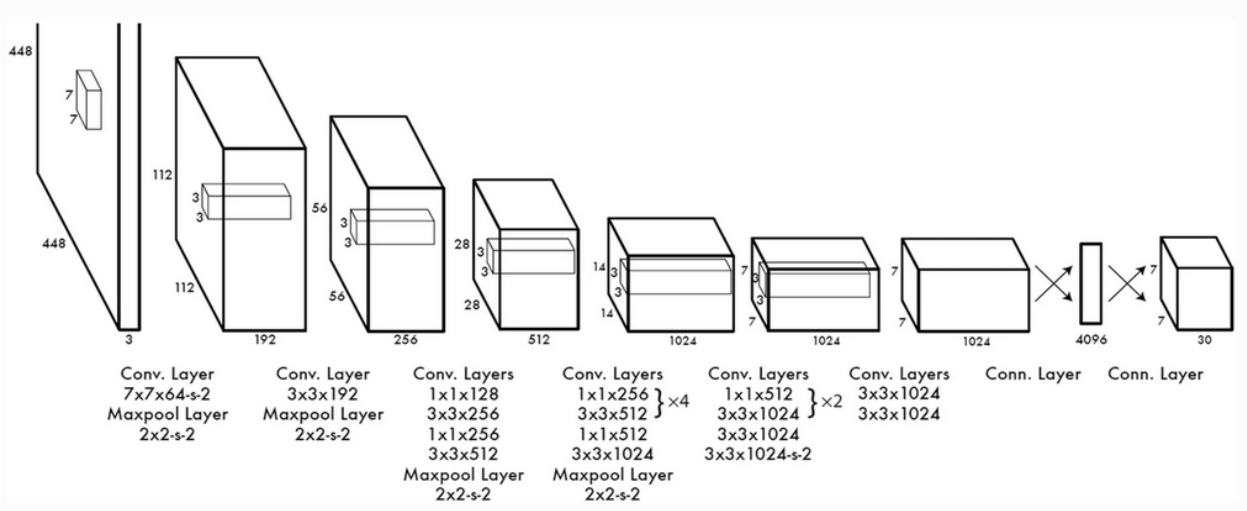


Figure 1.6 YOLO Architecture [37]

There are drawbacks in the first YOLO model such as low recall and localization errors compared to the two stage detectors that future versions of YOLO aim to improve. Fast YOLO is implemented with less complexity than the original YOLO architecture and has faster inference

but lower accuracy. Further improvements are made progressively over the years in each new version of YOLO with some limitations of their own. Table 1.1 shows the YOLO versions up to YOLOv5 with the architecture backbone, neck, head, loss function, and overall improvements the architecture addresses. The scope of this paper lies in the use of YOLOv4 and YOLOv5 which uses Path Aggregation Network (PAN) and Spatial Pyramid Pooling (SPP) as the neck to help extract features at different scales as well as the architecture being based on CSPDarknet53 that uses cross-stage partial connections along with the 53 layers. The YOLO model is a great option for the use in real time projects as the YOLOv4 is seen to reach 65 fps [38].

	Backbone	Neck	Head	Loss Function	Improvements
yolov1	GoogleNet	/	Fc→7×7×(5+5+20)	MSE	Directly fit the location of the Bounding Box
yolov2	Darknet19	/	Conv→13×13×5×(5+20)	MSE	Batch Normalization High resolution classifier Convolutional with Anchor Dimension clusters Direct location prediction Fine-Grained Features Multi-scale Training Hierarchical classification
yolov3	Darknet53	FPN	Conv→13×13×5×(5+80) →26×26×5×(5+80) →52×52×5×(5+80)	MSE	Multi-scale prediction Better classification network Binary cross-entropy loss
yolov4	CSPDarknet53	SPP+PAN	Conv→13×13×5×(5+80) →26×26×5×(5+80) →52×52×5×(5+80)	CIOU	Mosaic for data enhancement Using multi-anchors for single ground truth Eliminate_grid sensitivity(sigmoid) MSE loss→GIOU loss→CIOU loss
yolov5	Focus CSP Darknet53	SPP+PAN	Conv→13×13×5×(5+80) →26×26×5×(5+80) →52×52×5×(5+80)	GIOU	Adaptive anchor strategy Adopt Focus structure, CSP structure

Table 1.1 Yolo Differences

1.3.2 Instance Segmentation

Instance segmentation involves extracting the full shape of the object within the frame as compared to the bounding box around the object which does not give the best estimate of the size for the object. To perform instance segmentation, an algorithm must first locate all objects within an image and then create a separate mask for each individual

object. These masks are used to separate the objects from the background and from each other, allowing the algorithm to identify and classify each object individually. Each pixel with the bounding area of the object is given the label of the object which gives the instance segmentation models an advantage at the pixel level detections compared to standard object detection techniques with bounding boxes [39].

Facebook AI Research's software system Detectron is used for this project which starts with the Mask RCNN foundation benchmark. Segmentation algorithms are separated into two groups depending on if they rely on a region proposal system or do not. An example of an algorithm that does not use region proposal is the U-Net, which uses an encoder-decoder framework. The Mask-RCNN model uses a region proposal algorithm [40]. The Mask RCNN is an expansion of the Faster R-CNN, which has two primary stages. The first stage is the region proposal network, which spans over the image to locate regions that most likely contain objects. The second stage is the extraction stage, which extracts features from the proposed regions from the first stage. The extraction stage uses RoIPool from each possible box and performs the task of classification and regression. The Mask RCNN adds to the second stage, which includes predicting the binary mask for each RoI in parallel with the class and box offset. Mask RCNN uses four convolutional layers and two deconvolutional layers and no pooling layers within the head. It uses ResNet-101 as the backbone [41]. This model runs at 5 fps which is not suitable for the best real time response of initiating fast Raman scans.

1.3.3 Research into Microplastic Detection

Work by Lorenzo-Navarro, Javier, *et al*. [42] and V Wegmayr *et al*. [43] display the use of instance segmentation for microplastics using U-Net and VGG16. The use of a neural network

classifying holographic images of microplastics is used by Y Zhu *et al.* [44] and V Bianco *et al.* [45]. Holograms provide more information compared to the 2D images with the phase information of the objects included [38, 39]. Deeper feature extraction network structures are implemented because of the added information of the holographic images. This proposed system offers good performance in classification accuracy; however, the use of holographic images give difficulty in pretreatment, image reconstruction, and denoising. Another example of microplastic detection using machine learning is used by Z Chaczko *et a. l*[46]. In this research, hyperspectral imaging is used with a dataset of sixty textile samples with different texture patterns that are cropped to gain a larger dataset. The hyperspectral images give more information as well by analyzing a wider spectrum of light instead of assigning a value to the standard red, green, and blue channels. The texture patterns are used because there is no standard open-source dataset for microplastics which is a primary issue with microplastic analysis using computer vision. The neural network that is used in this work is created using Tensorflow's sequential model functionality, similar to how the Raman spectrum neural network is created in this thesis. The objects are first located using the find contours function in the grayscale image and then resized images based on the bounding area of the objects are created and then provided to the neural network in place. This provides a great example that uses hyperspectral images with neural networks that can be progressed using pictures of microplastics in the future when an open-source dataset is provided. These methods would require special sensors and the throughput for each is greatly diminished by using more data with these image types. The work in this thesis focuses on real time detection, therefore the standard three channel color images are collected.

1.4 Chemical Identification

Another method of analysis is chemical analysis, which involves using spectroscopy to determine the type of plastic present in a sample. Plastics are polymers that consist of long chains of repeating units called monomers. The consisting monomers within, how they are processed and polymerized will give a distinct chemical composition that can be classified. Polymers consist of polystyrene (PS), polyethylene (PE), polyethylene terephthalate (PET), polypropylene (PP), and polyamide (PA) as their prime raw materials. Spectroscopic analysis involves using spectroscopy techniques, such as Fourier-transform infrared spectroscopy (FTIR), Raman spectroscopy, or mass spectrometry to identify the chemical composition of microplastics. FTIR measures the plastic absorption or transmission of infrared radiation. Mass spectrometry measures the mass-to-charge ratio of ions produced when a plastic sample is ionized. This identification method is typically more precise than microscopic examination, as it can identify the specific type of plastic present in a sample. Chemical identification allows for a better understanding of what plastic products are entering the sampling location.

1.4.1 Raman Spectroscopy

Raman spectroscopy is a spectroscopic technique that is used to study the vibrational, rotational, and other low-frequency modes of a system. It is non-destructive and is widely used in chemistry, biology, and materials science. Raman spectroscopy is based on the Raman effect, which is the inelastic scattering of photons by a system. When a system is illuminated with a beam of light, some of the photons will be scattered by the system, and the scattered photons will have different energies and wavelengths than the incident photons. This shift in energy and wavelength is characteristic of the vibrational and rotational modes of the system, and it can be measured using a Raman spectrometer. The laser wavelengths that are available for Raman

spectroscopy are between 500 and 800 nm. Raman spectroscopy is a powerful tool for studying the chemical composition, structure, local environment, and dynamics for a range of areas including molecules, crystals, biological tissues, and plastic polymers which possess characteristic Raman spectra as well as the c. The use of Raman spectroscopy has seen successful results identifying microplastics with high reliability [47, 48]. A figure displaying Raman scattering and the schematic of the Raman spectroscopic methodology is shown in Figure 1.7

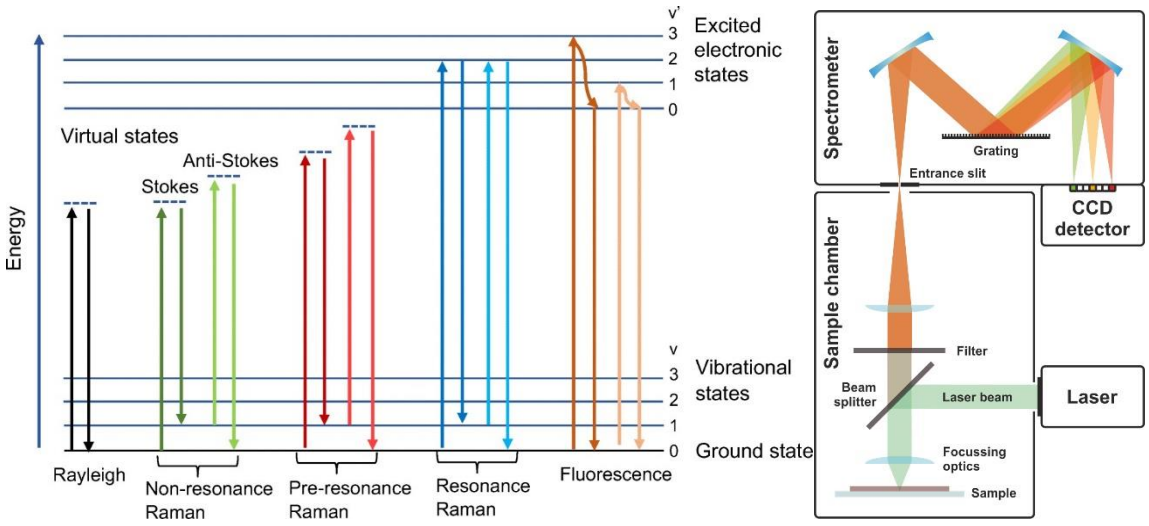


Figure 1.7 (a) Rayleigh scattering, Raman scattering, and fluorescence energy level diagram [49]; (b) Raman spectroscopy schematic [50]

Advantages of Raman spectroscopy and FT-IR over methods such as DSC or Pyr-GC-MS are that it is non-destructive, has a low sample requirement, a possibility for high throughput screening, and environmental friendliness [51]. The advantages of Raman spectroscopy over FTIR spectroscopy are that it shows better spatial resolution (down to 1 um in µRaman) which becomes relevant when identifying microplastics less than 20 µm, has a wider spectral coverage, higher sensitivity to non-polar functional groups, lower water interference, and narrower spectral

bands [51]. The negatives of Raman are that it has fluorescence interference, a low signal to noise ratio, and can cause damage and heating to the sample from the laser source which can lead to background emission.

1.5 Motivation and Overall Thesis Outline

The motivation of this thesis is to expand upon the portable system that is currently in place for sorting, detecting, and classifying microplastics. As presented in the introduction of the literature review, there is a need in research for the analysis of microplastics as more are spread throughout our world that impact all of life. There is not a standard practice for microplastic analysis, and this research hopes to provide a standard method for automated analysis of microplastics either in the field or in the lab that can remove laborious work. There is a difficulty in developing an efficient altogether system for microplastic analysis without the help of a team however because of the multiple factors that can affect the research such as weather, debris (dirt clogging the system), geographical location of sampling and time taken to sample in the field, time taken for data collection for the Raman scan library and the object detection library of images as there is no standard open source library of microplastic images or Raman scans as of date, throughput and timing of detection coupled with the Raman scanning without program delay, and efficiency of filtering and sorting methods. The work that is shown in this thesis is meant to provide an initial foundation for a system that can sort, detect, and classify microplastics that will be continuously improved upon. This project incorporates the methods listed in Table 1.2 below.

Mechanical Filtration	Use of strainers and a density sorting cyclone device to filter microplastics into the system
Computer Vision	Use of YOLO and instance segmentation to detect different morphologies of microplastics such as fibers, pellets, fragments, and a general category of dirt as well as gathering the sizing information of the objects, total count of objects, and to coordinate the timing of Raman scanning
Chemical Identification	Use of Raman spectroscopy to identify the chemical makeup of the microplastics: PET, HDPE, PVC, LDPE, PP, and PS. The #7 Other category will hopefully be implemented in the future of the project with more samples.

Table 1.2 Methodologies Incorporated in this System for Microplastic Analysis

The outline of this thesis is as follows. Chapter 2 presents the general system overview by first presenting the deployed system as a whole and then iterating through each induvial part from the input to the output. Chapter 3 describes the programming portion of the project in more detail with how the system operates. Chapter 4 presents the findings of the current system such as the neural network accuracies, counting, scanning efficiency, and field test results. There is still data and efficiency results needed to progress the project forward such as filtration and sorting statistics of the current setup with the cyclone and strainers at different depths of water, more experiments with testing the timing of Raman scans, more count and detection experiments with the colored neural network models, more field testing at different locations, and overall more data for the Raman and object detection libraries to improve accuracy. Chapter 5 presents

issues with the current project and improvements that can be made for future researchers to work on as well as the conclusion.

References

[1] Law, Kara Lavender, and Richard C. Thompson. "Microplastics in the seas." *Science* 345.6193 (2014): 144-145.

[2] Thompson, Richard C. "Microplastics in the marine environment: sources, consequences and solutions." *Marine anthropogenic litter* (2015): 185-200.

[3] Plastics—The Facts 2019. An Analysis of European Latest Plastics Production, Demand and Waste Data PlasticsEurope(2019)

[4] Eriksen, Marcus, et al. "Plastic pollution in the world's oceans: more than 5 trillion plastic pieces weighing over 250,000 tons afloat at sea." *PloS one* 9.12 (2014): e111913.

[5] Pagter, Elena, João Frias, and Róisín Nash. "Microplastics in Galway Bay: A comparison of sampling and separation methods." *Marine pollution bulletin* 135 (2018): 932-940.

[6] Jambeck, Jenna R., et al. "Plastic waste inputs from land into the ocean." *Science* 347.6223 (2015): 768-771.

[7] Galloway, Tamara S., and Ceri N. Lewis. "Marine microplastics spell big problems for future generations." *Proceedings of the national academy of sciences* 113.9 (2016): 2331-2333.

[8] Güven, Olgaç, et al. "Microplastic litter composition of the Turkish territorial waters of the Mediterranean Sea, and its occurrence in the gastrointestinal tract of fish." *Environmental pollution* 223 (2017): 286-294.

[9] Collard, France, et al. "Microplastics in livers of European anchovies (Engraulis encrasicolus, L.)." *Environmental pollution* 229 (2017): 1000-1005.

[10] Barboza, Luís Gabriel Antão, et al. "Marine microplastic debris: An emerging issue for food security, food safety and human health." *Marine pollution bulletin* 133 (2018): 336-348.

[11] Liebezeit, Gerd, and Elisabeth Liebezeit. "Synthetic particles as contaminants in German beers." *Food Additives & Contaminants: Part A* 31.9 (2014): 1574-1578.

[12] Kosuth, Mary, Sherri A. Mason, and Elizabeth V. Wattenberg. "Anthropogenic contamination of tap water, beer, and sea salt." *PloS one* 13.4 (2018): e0194970.

[13] Liebezeit, Gerd, and Elisabeth Liebezeit. "Non-pollen particulates in honey and sugar." *Food Additives & Contaminants: Part A* 30.12 (2013): 2136-2140.

[14] Yang, Dongqi, et al. "Microplastic pollution in table salts from China." *Environmental science & technology* 49.22 (2015): 13622-13627.

[15] Cox, Kieran D., et al. "Human consumption of microplastics." *Environmental science & technology* 53.12 (2019): 7068-7074.

[16] Ragusa, Antonio, et al. "Plasticenta: First evidence of microplastics in human placenta." *Environment international* 146 (2021): 106274.

[17] Wang, Chunhui, Jian Zhao, and Baoshan Xing. "Environmental source, fate, and toxicity of microplastics." *Journal of hazardous materials* 407 (2021): 124357.

[18] Zhao, Kai, et al. "Separation and characterization of microplastic and nanoplastic particles in marine environment." *Environmental Pollution* (2021)

[19] Wang, J., et al. "Runoff and discharge pathways of microplastics into freshwater ecosystems: A systematic review and meta-analysis." *FACETS* (2022).

[20] Eriksen, Marcus, et al. "Microplastic pollution in the surface waters of the Laurentian Great Lakes." *Marine pollution bulletin* 77.1-2 (2013): 177-182.

[21] Mason, Sherri A., et al. "Pelagic plastic pollution within the surface waters of Lake Michigan, USA." *Journal of Great Lakes Research* 42.4 (2016): 753-759..

[22] Moore, Charles James, Gwendolyn L. Lattin, and A. F. Zellers. "Quantity and type of plastic debris flowing from two urban rivers to coastal waters and beaches of Southern California." *Revista de Gestão Costeira Integrada-Journal of Integrated Coastal Zone Management* 11.1 (2011): 65-73.

[23] Isabel Jalon-Rojas, Erick Fredj, Xiao Hua Wang, "Numerical Model Simulation to improve understanding of Micro-Plastics Debris in marine environments: sensitivity of microplastics fate to particles physical properties and behavior", MICRO 2018, Fate and Impact of Microplastics: Knowledge, Actions and Solutions. Lanzarote, 19-23 (2018)

[24] Weiss, Lisa, Wolfgang Ludwig, and Claude Estournel. "Modeling plastic waste flows in the Mediterranean environment: a source to sink approach." *MICRO 2018. Fate and Impact of Microplastics: Knowledge, Actions and Solutions*. MSFS-RBLZ, 2018.

[25] Razeghi, Nastaran, et al. "Microplastic sampling techniques in freshwaters and sediments: a review." *Environmental Chemistry Letters* 19.6 (2021): 4225-4252.

[26] Eriksen, M., Mason, S., Wilson, S., Box, C., Zellers, A., Edwards, W., et al., Microplastic pollution in the surface waters of the Laurentian Great Lakes. *Marine Pollution Bulletin, 77*(1–2), 177–182 (2013)

[27] Morét-Ferguson, Skye, et al. "The size, mass, and composition of plastic debris in the western North Atlantic Ocean." *Marine Pollution Bulletin* 60.10 (2010): 1873-1878.

[28] Shen, Maocai, et al. "Removal of microplastics from wastewater with aluminosilicate filter media and their surfactant-modified products: Performance, mechanism and utilization." *Chemical Engineering Journal* 421 (2021): 129918.

[29] Mansouri, Borhan, et al. "Bioaccumulation and elimination rate of cobalt in Capoeta fusca under controlled conditions." *Chemical Speciation & Bioavailability* 25.1 (2013): 52-56.

[30] Zhao, Kai, et al. "Separation and characterization of microplastic and nanoplastic particles in marine environment." *Environmental Pollution* (2021)

[31] Salafi, Thoriq, Kerwin Kwek Zeming, and Yong Zhang. "Advancements in microfluidics for nanoparticle separation." *Lab on a Chip* 17.1 (2017)

[32] Gent, Malcolm Richard, et al. "Recycling of plastic waste by density separation: prospects for optimization." *Waste management & research* 27.2 (2009): 175-187

[33] Fok, Lincoln, et al. "A meta-analysis of methodologies adopted by microplastic studies in China." *Science of the Total Environment* 718 (2020): 135371.

[34] Borgia, Fabio. "Performance Analysis and Modeling of Microplastic Separation through Hydro Cyclones." *Waste Material Recycling in the Circular Economy-Challenges and Developments*. IntechOpen, 2021

[35] Zou, Zhengxia, et al. "Object detection in 20 years: A survey." *arXiv preprint arXiv:1905.05055* (2019).

[36] Bengio, Yoshua, Aaron C. Courville, and Pascal Vincent. "Unsupervised feature learning and deep learning: A review and new perspectives." *CoRR, abs/1206.5538* 1.2665 (2012): 2012.

[37] Diwan, Tausif, G. Anirudh, and Jitendra V. Tembhurne. "Object detection using YOLO: Challenges, architectural successors, datasets and applications." *Multimedia Tools and Applications* (2022): 1-33.

[38] Bochkovskiy, Alexey, Chien-Yao Wang, and Hong-Yuan Mark Liao. "Yolov4: Optimal speed and accuracy of object detection." *arXiv preprint arXiv:2004.10934* (2020).

[39] Jiang, Peiyuan, et al. "A Review of Yolo algorithm developments." *Procedia Computer Science* 199 (2022): 1066-1073.

[40] Redmon, Joseph, et al. "You only look once: Unified, real-time object detection." *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2016.

[41] Ojha, Apoorva, Satya Prakash Sahu, and Deepak Kumar Dewangan. "Vehicle detection through instance segmentation using mask R-CNN for intelligent vehicle system." *2021 5th international conference on intelligent computing and control systems (ICICCS)*. IEEE, 2021.

[42] Johnson, Jeremiah W. "Adapting mask-rcnn for automatic nucleus segmentation." *arXiv preprint arXiv:1805.00500* (2018).

[43] He, Kaiming, et al. "Mask r-cnn." *Proceedings of the IEEE international conference on computer vision*. 2017.

[44] Lorenzo-Navarro, Javier, et al. "Deep learning approach for automatic microplastics counting and classification." *Science of the Total Environment* 765 (2021): 142728.

[45] Wegmayr, Viktor, et al. "Instance segmentation for the quantification of microplastic fiber images." *Proceedings of the IEEE/CVF winter conference on applications of computer vision*. 2020.

[46] Zhu, Yanmin, Chok Hang Yeung, and Edmund Y. Lam. "Microplastic pollution monitoring with holographic classification and deep learning." *Journal of Physics: Photonics* 3.2 (2021): 024013.

[47] Bianco, Vittorio, et al. "Microplastic identification via holographic imaging and machine learning." *Advanced Intelligent Systems* 2.2 (2020): 1900153.

[48] Goodman, Joseph W. *Introduction to Fourier optics*. Roberts and Company publishers, 2005.

[49] Ren, Zhenbo, Tianjiao Zeng, and Edmund Y. Lam. "Digital holographic imaging via deep learning." *Computational Optical Sensing and Imaging*. Optica Publishing Group, 2019.

[50] Auner, Gregory W., et al. "Applications of Raman spectroscopy in cancer diagnosis." *Cancer and Metastasis Reviews* 37 (2018): 691-717.

[51] Van Cauwenberghe, Lisbeth, et al. "Microplastic pollution in deep-sea sediments." *Environmental pollution* 182 (2013): 495-499.

[52] Cole, Matthew, et al. "Microplastic ingestion by zooplankton." *Environmental science & technology* 47.12 (2013): 6646-6655.

[53] Schmid, Thomas, and Petra Dariz. "Raman microspectroscopic imaging of binder remnants in historical mortars reveals processing conditions." *Heritage* 2.2 (2019): 1662-1683.

[54] Elert, Anna M., et al. "Comparison of different methods for MP detection: what can we learn from them, and why asking the right question before measurements matters?." *Environmental Pollution* 231 (2017): 1256-1264.

CHAPTER 2 SYSTEM OVERVIEW

2.1 Deployed System

The current overall deployed system is first presented and then it will be broken down into its smaller components that allow the functionality of sorting, detecting, and classifying microplastics. The system was designed to be easily transported to testing locations, with the goal of making it as small as possible in future revisions. The initial system used mechanical filtration, YOLO for object detection, and a Raman spectrometer for classification. The current setup has a steel cart with the devices attached or drilled in place. An example of the system in a field test could be seen in Figure 2.1, where each side of the cart is represented.

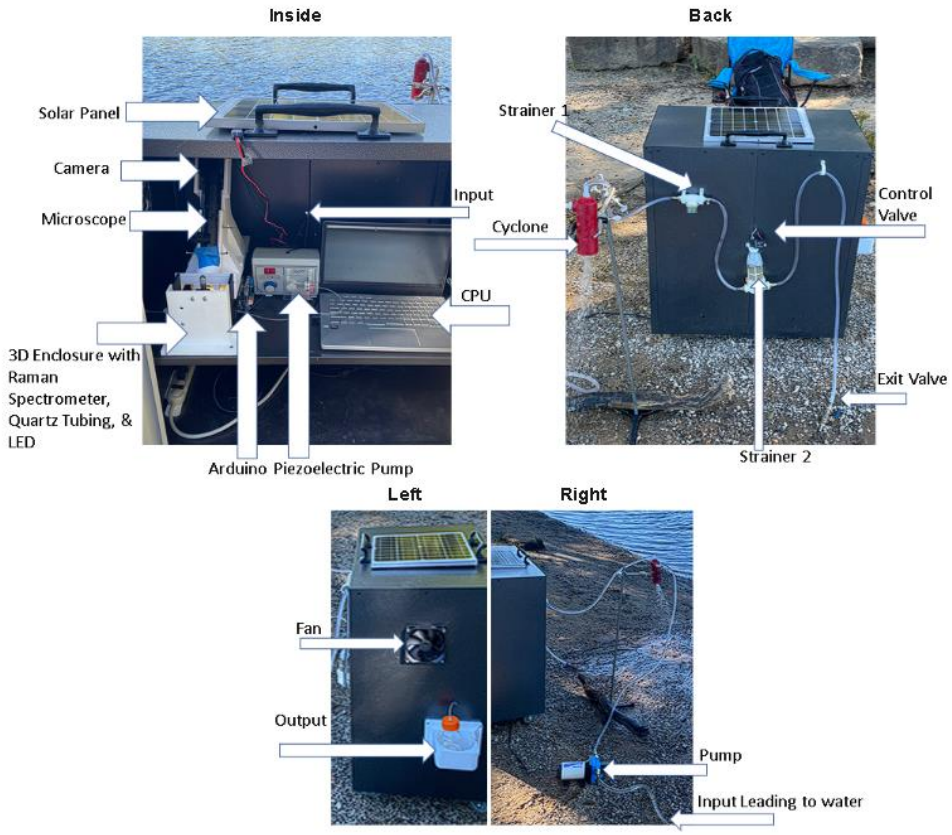


Figure 2.1 Deployed System View

The current system makes it easy to set up for field tests. To do so, the researcher needs to roll the cart to the location from the car, along with the chair, pump, cyclone, and cyclone stand. Additional items, such as 3D-printed handles for strainers and hoses, tubing, scissors, and zip ties, may be brought to the field tests if an issue occurs. Testing has primarily been done using extension cords that can reach up to 150 feet from the testing site to the power outlet. In some cases, a portable power supply or small solar panel has been used to power the devices, but the portable power has not been fully incorporated yet. Using an outlet has proven to be reliable for the sites that have been tested so far with longer periods of testing, but for more remote locations, a more suitable power supply will be used in the future to enhance portability. Once the cart is in position at the site, the researcher needs to set up the

cyclone with the stand, connect the external pump to the power outlet within the cart, connect their laptop inside the cart, and connect the power source to start.

The stainless-steel cart that houses all the devices, strainers, and output has two doors that can be locked with a key. This is useful for testing in certain conditions or sites where the testing period will continue overnight and the devices within the cart will be secured. The cart measures 35.43 inches in height, 30.31 inches in length, and 18.11 inches in width. This size allows the cart to fit in the back of an SUV when laid down on its back. The cart has a drawer that separates the main devices at the top from the power cables, power supply, PCI adapter, or external hard drive at the bottom. The total width of the control pump, 3D enclosure with an Arduino drilled at the side, and laptop is twenty-seven inches in total. The wires are grouped with zip ties and run through the slot between the drawer and the back of the cart to the devices. The 3D-printed enclosure, pump, adjustable power supply, PCI adapter, and multi-port outlet are secured to the stainless-steel using screws and command strips. A hole is drilled through the bottom of the cart to connect the multi-port outlet to the external pump and wall outlet. A large hole is drilled for the fan on the upper left side of the cart, and two small holes are drilled for the input and output to the system within the cart. The input hole, with a diameter of 1cm, is located in the middle of the backside above the control valve where the 1/16" hose and power cables for the valve pass through. The output hole, with a diameter of 0.5cm, is in the middle of the left side where the 1/16" hose leads to the output container connected outside the cart. 3D-printed holders are drilled into the backside of the cart to hold the external strainers and tubing, and a 3D-printed piece is drilled into the left side to hold the output container.

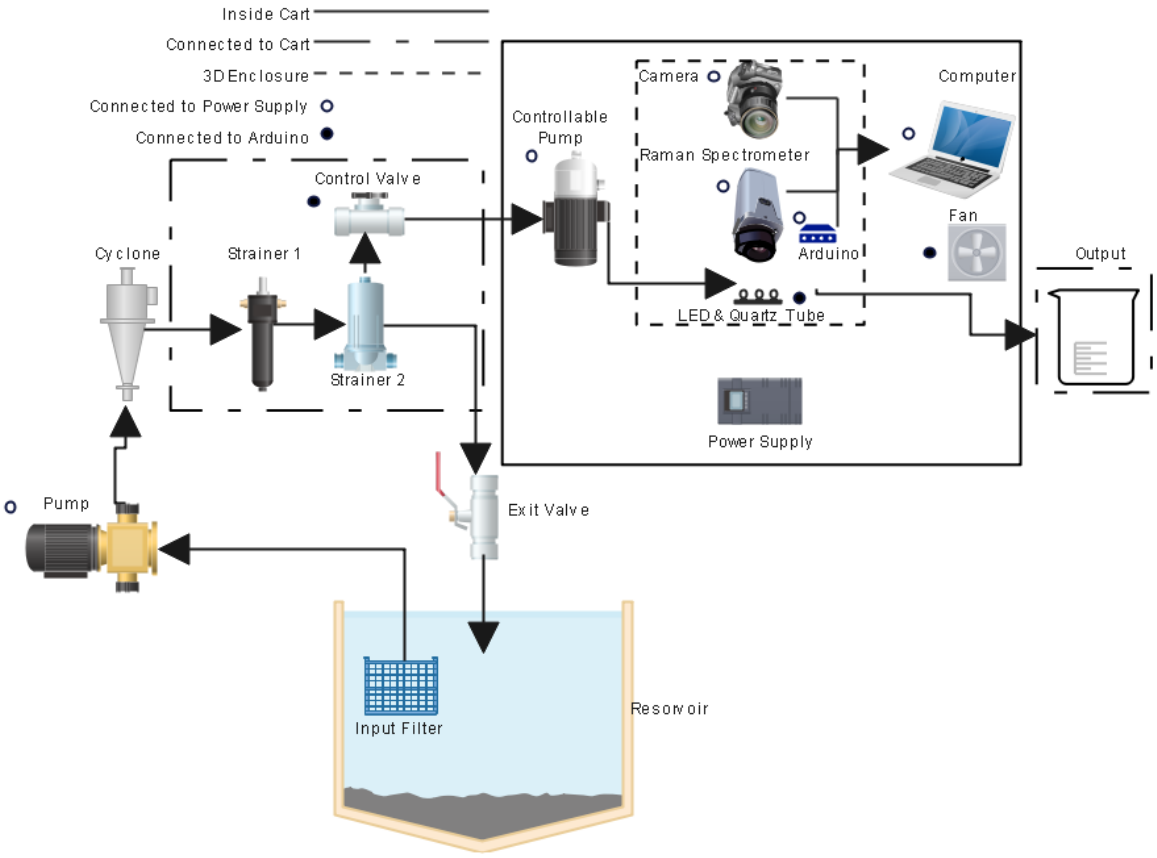


Figure 2.2 System Flow Chart

In the next subsections, the system is broken down and each device used in the system will be explained based off Figure 2.2 which displays the flow of the sample through the system and every device that is used. A legend can be seen in the upper left which displays what each line and dot is for within the diagram. The solid line is for each item that is inside of the cart, the partially dashed line is for each item that is connected externally to the cart, and the fully dashed line is used to represent what is within the 3D printed enclosure. The open dot represents each item that is connected to the power supply and the closed dot represents each item that is connected to the Arduino. The parts separate from the cart are without bounding boxes and currently lie on the ground or with a stand.

2.2 Filtering

The filtering section includes the items connected to the outside of the cart within the partially dashed lines and the ones separate from the cart without a bounding box. The mini pump inside of the cart is incorporated in this section as well. The use of mechanical filtration is the prime method for sampling microplastics in this research and there are several advantages to using filtration for microplastic sampling. It is relatively simple and cost-effective and can be used to sample a wide range of matrices including surface water, groundwater, and seawater. It is also relatively quick, with samples typically being collected in a few hours or less based on the flow rate.

However, there are also some limitations to this method. One major limitation is that it is limited to particles that are larger than the pore size of the filter. This means that smaller microplastics may not be captured, leading to an underestimate of their abundance. In addition, the efficiency of the method can be affected by factors such as the flow rate of the water, the size and shape of the microplastics, and the presence of other particles such as dirt or algae that may clog the filters.

The size range for filtering is mainly dependent on the accuracy and consistency of the Raman scan spectrums and positioning for the scan within the viewing tube. Smaller particles are much more difficult to position at the scan location within a tube where the water is continuously flowing. The large scan area gives a limitation on the size of the objects that will pass through the system that can accurately and consistently be scanned. The size range of the microplastics chosen is primarily 100-500um. There are two strainers and an initial mesh at the input that sort based off size and a 3D printed cyclone is used for sorting based off density.

2.2.1 Input

Starting at the input, the entrance sieve connected to the hosing had a 5mm mesh size and was submerged into the water close to surface level. Resources at the testing site such as large rocks within the water and buoys have been used to hold the position of the input in place and near the surface level of the water as the incoming waves come in. This idea comes from the notion that the density of most microplastics is within the same range of density as water and with the testing primarily done at the shoreline, the depth of water is close to two feet and a majority of microplastics will be suspended close to the surface of the water as incoming waves come in or at the surface itself. A representation of the plastic density is shown in Figure 2.3. In reference to water which has a density of 1 g/cm^{-3}, microplastics that are denser such as those made with PVC or PET could be more difficult to be acquired in this filtering strategy because particles above this level of density would sink and either lie at the bed of the water or be suspended further below the surface.

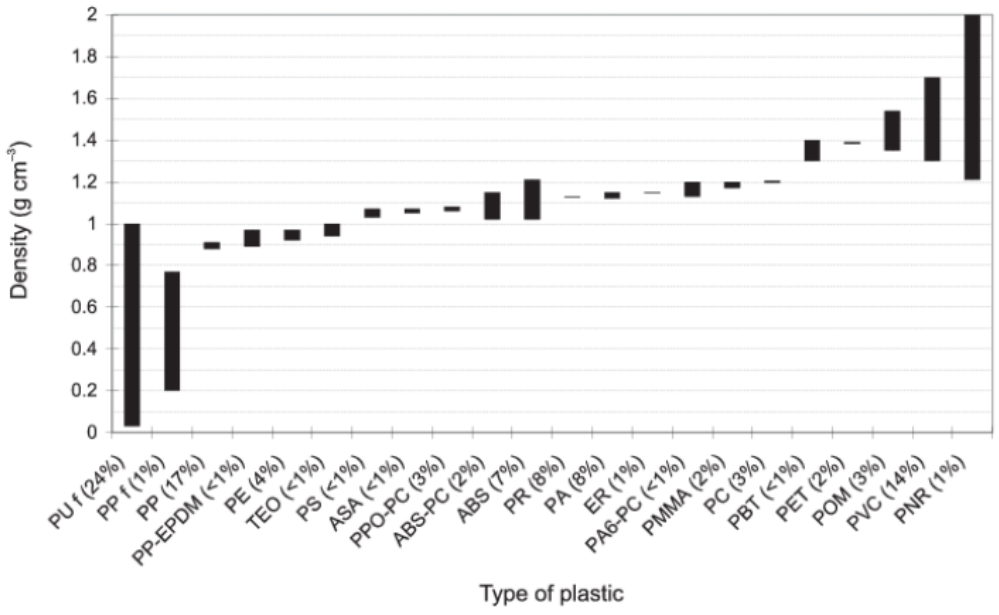


Figure 2.3 Plastic Density Ranges with Percent Accumulation Present in a 1995 Family Sedan (from Plastics International 2007) [1]

The hosing that was attached to the entrance filter was 3/8”. This input has limitations in that the diameter is small and can only sample from the one area. Another limitation of the input is that it only samples at one position within the depth and location of the water. More techniques are discussed in chapter five on possible inputs that are being tested.

2.2.2 Pumps

The system has two pumps. The input was connected to the main external pump that drives the water through the system. The pump is a DC HOUSE branded 42-Series Upgrade Water Diaphragm Pressure Pump. It pumps five gallons per minute (GPM) at fifty-five PSI with twelve volts direct current and seven amps. It is self-priming and has a built-in pressure switch that ranges from 40-80 PSI as well as a temperature control protector.

The second pump is a VWR branded Variable-Speed Peristaltic Pump that is within the cart and takes in input from the control valve outside of the cart and traverses the sample through the 3D designed enclosure (where the object detection and Raman scanning is initiated) and then to the output. This pump has ninety-nine settings that can be manually switched using a knob. It has the 1/16” tubing size. The pump can flow from 57.4 to 615.4 µl/min within the ninety-nine settings with the current amount of hosing connected. Flow rate results can be seen in Table 4.1.

2.2.3 Cyclone

The external pump was connected to the 3D printed cyclone that was held with a stand. Examples of the cyclone designs are in Figure 2.4. These cyclone designs were designed based off the literature within section 1.2.1. The top and bottom cone designs were

made separately for the 3D prints to succeed without the pathing being filled in during the printing job.

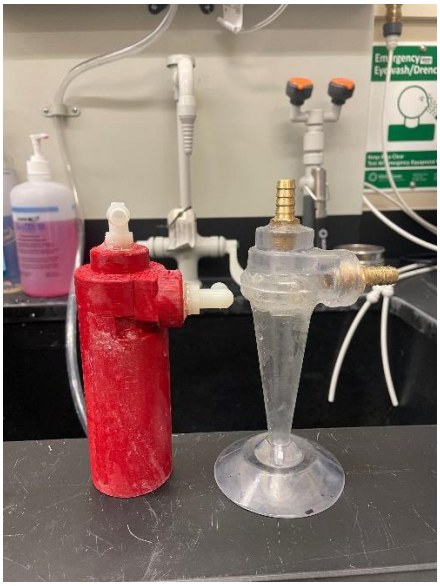


Figure 2.4 Cyclone Designs

The cone design on the right was printed in a clear resin for the use of viewing the process of the cyclone sorting the microplastics by density using a high-speed camera within the lab. The design on the left was printed with a colored PLA and with more support material for rigidity in the field tests. The PLA is better for sealing using methylene and does not get affected by the UV light from the sun that affects the resin-based prints. Tiny holes within the PLA print are the main concern as water can escape when the pressure builds within. Applying glue, methylene, or wrapping material around the unit helps with eliminating these possible leaks.

As explained in section 1.2.1 of this thesis, the input to the cyclone is from the side and the output is at the top, where the sample will continue through the system. The denser particles will pass through the bottom exit and back into the environment while the lighter particles will then traverse up through the output and to the strainers.

2.2.4 Strainers

The first strainer labeled as strainer 1 in Figure 2.2 has a 300-um mesh to further sort out any larger particles that might have passed through from entering the system. The second strainer labeled as strainer 2 has a 100-µm mesh and was much larger to better contain more particles during longer periods of testing before having to clean the mesh. The size of the strainer is also a limitation as the strainer needs to be fully filled for the top opening to give input to the internal system inside of the cart. This strainer contains the particles that will be viewed inside of the cart. Strainer 2 has three inlets which can be viewed in the back view within Figure 2.1. The bottom two are the main input and exit of the strainer and the top inlet is the collected output of the strainer to the control valve and then to the system within the cart. The right exit of the strainer is the filtered-out water that will then flow to the exit valve and back into the environment. The 3/8” hose barbs need to be fastened as tight as possible into the strainers as well as using hose clamps attached to the hosing to diminish possibilities of leaking or the hosing disconnecting during the field tests under pressure overtime. The strainers were held up by 3D printed handles that are drilled into the wall of the cart. A limitation to these strainers is the buildup of particles within the mesh which clog and provide more pressure to the system if not alleviated. This buildup is diminished by using multiple strainers and having the majority of the denser particles passing out through the cyclone. The input position of the hose is also vital for diminishing buildup, as the closer the input gets to the bed of the testing area, the more dirt particles will pass through at a time.

2.2.5 Valves

There are two valves that were used in the system. There was a 12V 400mA control valve with 1/4” threading connected to a 1/4” to 3/8” adapter to strainer 2. Its power connections are connected to the power shield of the Arduino, and it was controlled using the serial port. It was controlled to stop or start the flow to the system inside of the cart and will be used for future improvements for back flow and input control. There is a possibility of rising pressure to the system that can cause inner hosing to be disconnected within the cart that can damage components and having this control valve to switch off the flow if needed is helpful. The 1/4” to 1/16” hose barb is then connected from the valve to the 1/16” hosing that leads to the control pump within the cart.

The exit valve was connected to the exit of strainer 2 with the filtered sample that flows out to the environment. This exit valve had a handle that can be used to provide or alleviate pressure to the system which can help to speed up filling up strainer 2 to progress the flow to the internal system as well as alleviating pressure changes from large incoming waves that can affect the flow to the system.

2.2.6 Output

A 3D printed beaker holder was drilled into the left side of the cart seen in Figure 2.1. This can hold a variety of containers up to 150mm in width. A 250mL container was used for field testing which gets filled at a flow rate of 62.9 µl/min in 68.42 hours of field testing. That flow rate is referenced based on the results seen in chapter 4 from the Raman scanning tests. When the Raman scanning functionality was removed in field testing to gather more data of what was gathered, the flow rate to the output can increase to 615.4

μl/min with the highest setting of the mini pump which filled the output container in 6.77 hours of testing.

2.3 3D Printed Enclosure

The output of the controllable pump was the input to the 3D printed enclosure where the high-speed camera, microscope, Raman spectrometer, adjustable base, LED, quartz tube, and the holder for the LED and quartz tube was located within. An Arduino was drilled into the side of the enclosure as well. The function of the enclosure was to allow the viewing tube's location to be adjusted, for the Raman spectrometer and camera to be aligned at the same position of the tube, and to block out excess light that can interfere with the Raman spectrums. Subsections of 2.3 are created to describe each component connected within the enclosure.

The largest 3D printer on the facility that can be used by students is the Ultimaker S5, which has a build size of 330(w) x 240(h) x 300(d) mm. This limited the total size of prints that can be done. The full design of the printed enclosure was designed and then cut into separate pieces that the 3D printer can handle as well as allowing some freedom for the user to handle issues that might arise without having to take the whole piece apart. Iterations of the design have been made for an optical cabled light source and for two different cameras. There were six total pieces that make up the enclosure and they all are screwed together and can be separated or changed out with a new testing piece. There were screw hole locations at each bottom piece to be drilled into place inside of the cart. The fully printed piece with the components combined can be seen in Figure 2.5. The bottom displays the inside of the enclosure, and the top shows the exterior.

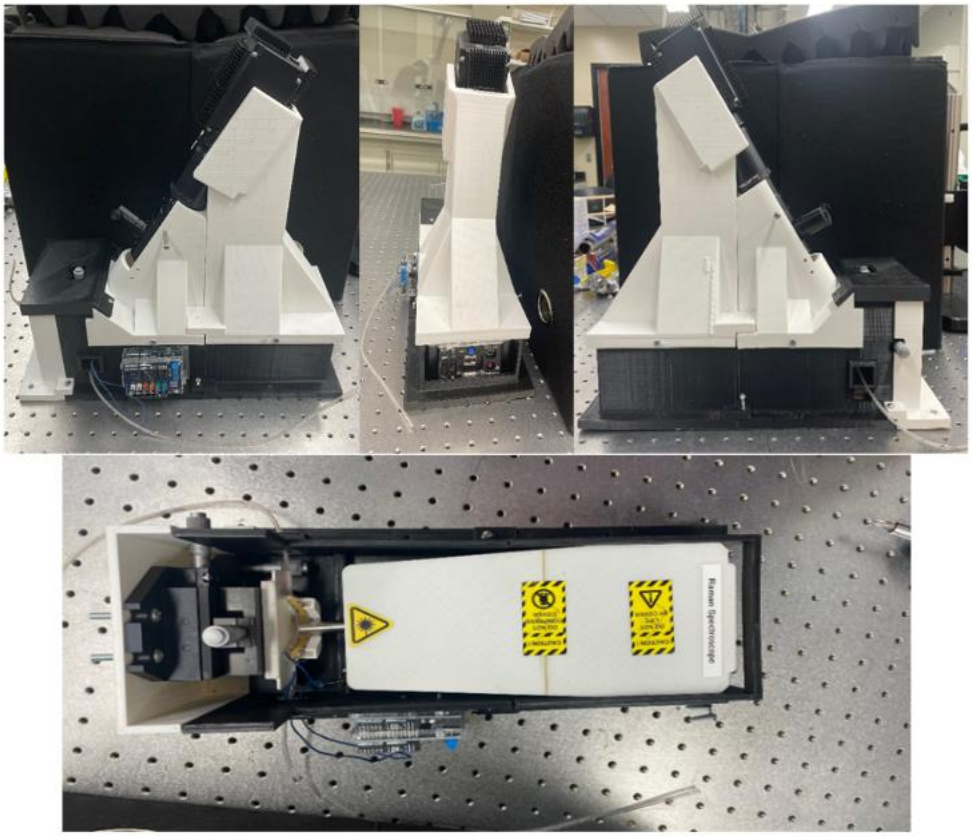


Figure 2.5 3D Printed Enclosure

2.3.1 3-axis Adjustable Stage

A base piece was drilled within the front of the enclosure that allows the user to adjust the x, y, and z directions using three knobs that can be accessed outside of the enclosure. The adjustable base allows the user to test at different lengths from the Raman spectrometer as well as different locations of the tube that need to be viewed with the camera. Each part was securely drilled together and the whole piece was drilled into the cart, but since the enclosure was made with multiple pieces and they were made with PLA, there still is the possibility for a disturbance to the cart that can shift an item within. Having this piece to adjust to those minor deviations was helpful. The vertical y direction of the piece can be adjusted using the top knob, the horizontal x direction can be adjusted with the upper

side knob that is extruded out of the enclosure, and the displaced z direction, which is the length of the base to the Raman spectrometer, was controlled by the lower side knob that is within the enclosure. A close-up image of the adjustable piece with the attachments included is in Figure 2.6.

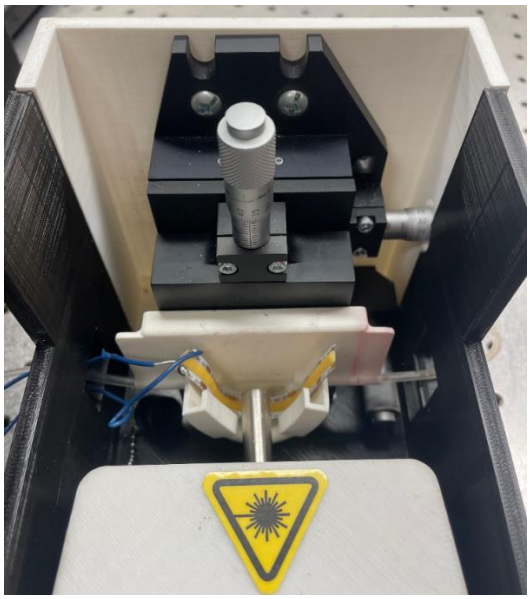


Figure 2.6 Stage, Holder, LED, & Quartz

2.3.2 Quartz. LED, & Holder

A 3D designed piece was designed to hold the viewing quartz tube in place on the middle of the adjustable base as well as the LED that is the light source to the tube. The input and output of the holder was designed to hold the 1/16” tubing in place that was connected over the quartz tube to give more security to the connections of the quartz which can easily be broken.

The quartz tube used had a length of 86mm and a width of 1.50mm. Quartz tubing was used because the material does not affect the Raman scan of the object within the tube as compared to glass tubing which gave undesired noise. The tubing within was chosen to be

rounded to allow better viewing for the camera at its current location and angle from the tube.

The LED was wrapped within the 3D printed holder. The power connections traversed through the input window of the enclosure to the Arduino that was drilled into the input side of the enclosure.

2.3.3 Raman Spectrometer

The spectrometer lied on the bottom of the enclosure and was positioned in front of the viewing tube and base. The length away from the base was determined by testing the accuracy of the scans at the center location of the z axis.

2.3.4 Camera & Lens

There are two primary cameras that have been used to test with the system. The first camera that has been tested is the Andor Zyla 5.5 sCMOS. This camera has a 5.5-megapixel resolution and offers a 100 fps Camera Link with ultra-low noise performance. With the right hardware and image parameters, this camera can reach over 1000 fps. The primary limitation is that it was only in grayscale and the connection was not compatible with laptops without a proper pci adapter.

To account for these limitations, a color camera was introduced. The second camera that is tested is the Lucid Atlas10 and is the same camera seen in Figure 2.5. It offers an 8.1 mega-pixel resolution with a total pixel size of 2840x2840. The camera can achieve a fps of 136.7 at and can be increased by adjusting the image parameters.

A ThorLabs 12X zoom lens with 3mm fine focus and an illumination port was used with the cameras. The large magnification can be used in future iterations of the project where the size range of the particles examined are reduced. The length of the lens from the viewing

tube can be adjusted with screw holes in the enclosure. Increasing the distance of this length gives focus to different levels of suspended objects within the tube as well as reducing the image size needed for the viewing tube. However, increasing the distance also reduced the clarity of features for the objects within view as the focus window gets smaller. This is a difficult tradeoff to deal with as for this project, the accuracy of determining objects is better with more distinct features, but every object passing through the tube at different depths also needs to be viewed and accounted for. A set position in the enclosure is made by testing different locations for the best view of the more suspended particles within the tube without taking away too much clarity on the full image.

2.4 Processing

The control and collection of data for each device was done using the researcher's laptop and an Arduino.

2.4.1 Arduino & Motor Shield

The Arduino Uno and Mega are microcontrollers that have been used to control the LED, fan, and control valve. An Arduino Uno operates at 16 MHz which means it can execute up to sixteen million instructions per second. The hardware and software are open source which was great for finding information to help with projects. It also has a low power consumption using the basis of the Microchip ATmega328P and the Alf and Vegard's RISC (AVR) architecture within the board. The AVR processor is one of the first microcontroller groups to use on-chip flash memory [2]. The Uno has 32 KB of flash memory and operates at 4.75 to 5.25 V using an USB connection to the computer. It is 68.6 by 53.4mm in size which is great for mounting to the 3D enclosure.

An Arduino motor shield REV3 was connected to the Arduino to account for the higher voltage needed for the devices. The motor shield is based on the L298 dual full-bridge driver that allows driving inductive loads such as stepping motors or relays with an operating voltage of 5V to 12V. This shield allows the user to drive two DC motors at a time for a total of two Amps per channel. The twelve-volt units are connected to the positive and negative pins of the shield while the LED is connected to the digital output pin of the shield.

2.4.2 Computer

The laptop that has been used primarily in field tests is the Asus ROG Zephyrus G14. This device has a 14-inch body weighing at 1.6kg. It has an 8-core Ryzen 9 4900HS CPU and sixteen threads. The GPU is a NVIDIA GeForce RTXTM. The spectrometer, camera, and Arduino is connected to the laptop. This computer operates at 180W.

2.5 Power & Externals

A three-to-twelve-volt adjustable power supply has been used for powering the components connected to the Arduino power shield (LED, fan, and control valve). This supply has the circuitry compacted into a 5 by 13.3 cm encasing which is in place at the back wall of the top shelf of the cart. A Buffalo branded quad DriveStation has been used at the bottom of the cart for external memory within the system for the possible situation of memory limits within the local computer. This station has four drives with four TB each.

References

[1] Gent, Malcolm Richard, et al. "Recycling of plastic waste by density separation: prospects for optimization." *Waste management & research* 27.2 (2009): 175-187.

[2] Banzi, Massimo, and Michael Shiloh. *Getting started with Arduino*. Maker Media, Inc., 2022

CHAPTER 3 PROGRAMMING

3.1 Main Program Workflow

The primary application written in C++ that controls the devices and collects the data needed is represented as the flowchart in Figure 3.1. Variations of the program have been made with Python and LabView as well that exclude the Raman classification and can use different methodologies for detecting objects.

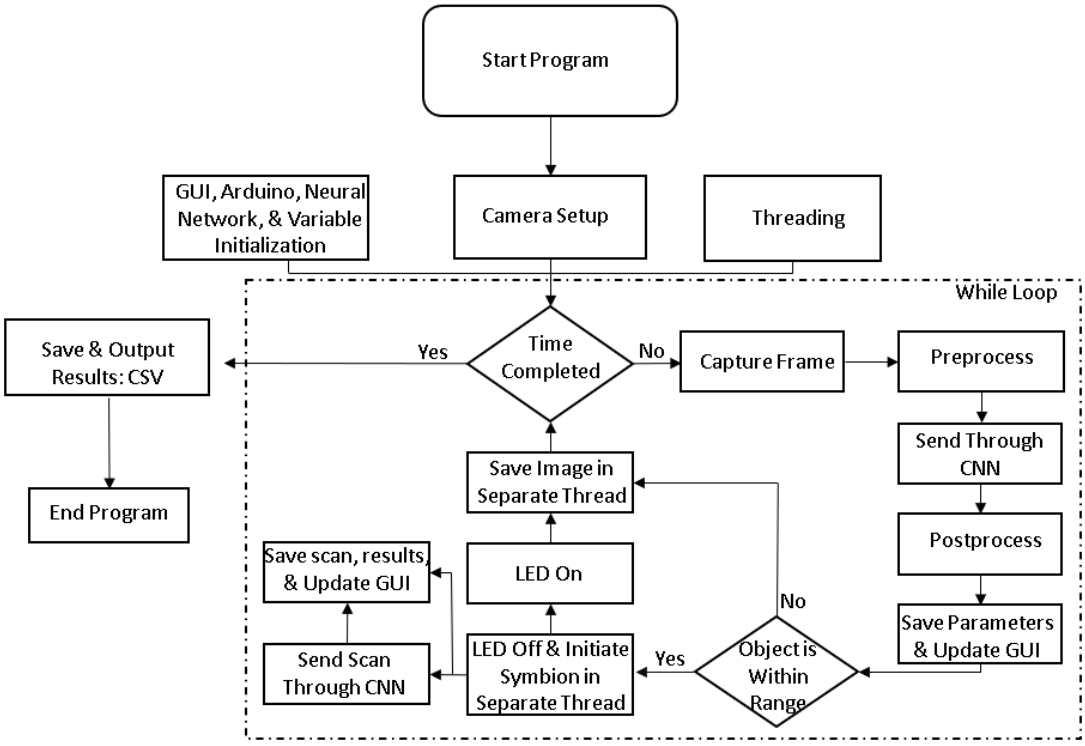


Figure 3.1 Program Flow

C++ was primarily used for its speed performance but lacks the flexibility and ease that other languages hold which gives a learning curve for complex programming tasks.

LabView and python also have interfaces and libraries that can easily be operated and are much more difficult to implement in C++. A primary goal in this program is to have a great amount of modularity where aspects can easily be changed or removed. Having flexibility with how the program is ran and making it modular can help future researchers progress the work.

Before the primary program can be started, the camera, spectrometer, and Arduino need to be on and connected to the computer. The spectrometer then needs to be activated within its driver software. The project is maintained and compiled through Visual Studio. Each connecting piece within the diagram will be explained in the upcoming subsections.

3.2 Initialization

Every variable, file, and library need to first be initialized and included within the program. The proper pathing to the neural network for object detection needs to be loaded in as well as parsing through the label file and creating the label vector. The backend for the network needs to be set to CUDA and the output layer names need to be acquired. Each variable array and proper output file locations need to be set for the results of the detected objects, Raman scans, video, or images that need to be saved.

3.2.1 GUI Setup

The statistics of the objects detected such as the label, size, and count as well as the scan count and scan label need to be presented and updated throughout the program for the researcher to view. Having a GUI allows for better visibility of results during testing. There is also an importance around the speed of the program in that the higher frame rate that is achieved, the more optimized the program can initiate a scan and collect data from what is detected so having a GUI that can be quickly updated is beneficial. Dear ImGui was the

primary graphical user interface library that was chosen which uses the immediate mode GUI paradigm. This library enables efficient runtime and memory consumption that minimizes state synchronization, state storage, and setup [1].

In the initial startup of the program, the user can set the time limit that needs to be executed and the camera width/height. The program will then operate until the time limit has been reached. Two bar charts can be viewed that display the count of each object detected as well as each scan type that is determined. A summary table for the count of each object detected, the average confidence score for that class, and the current fps was also displayed. An example of the display with generated data to give a more diverse representation is shown in Figure 3.2. There are improvements to be made with this GUI that are discussed in chapter 5.

Figure 3.2 GUI Representation

3.2.2 Arduino Script

A script was uploaded to the Arduino that switches the power for the control valve, LED, and fan using serial input. The baud rate was set to 115200 and it takes 8.68μs per bit to read in which allows 86.8μs for a character to be entered into the Arduino with stop and

start bits. All three devices are turned on at the beginning of the program. The control valve and LED are the primary devices that need to be switched on and off, whilst the fan stays on throughout the execution of the program. A MouseCallBack function was created with the image window where the program takes in mouse input while hovering over the current frame. When a user double clicked the window, the control valve will switch from on to off by use of a count and if statement checking for odd or even to not have to use two different mouse inputs. The control valve was used for the special case of an incident occurring within the cart and the user needing to stop the flow of water inside. The LED was turned off or on based on the detection location and type of the object to initiate Raman scanning so that there is no interference of light for the spectrum.

3.3 Threading

Multithreading is one of the most important aspects of the program as it alleviates the processing power from the main thread and allows functions or programs to be initiated concurrently. There were two separate threads that were initiated and called upon throughout the program. The main thread was the main loop through the program that took in each frame, preprocessed it, traversed it through the neural network, and then postprocessed it.

A saving thread was one separate thread that was used to save each frame when a video was needed to be acquired as an output. Each frame was concatenated to a vector that was sent to the separate thread and the image was saved to the file location and then deleted from the vector. A large drop in fps can be seen when attempting to save each frame without the use of threading.

A scanning thread was the other separate thread that was used to initiate the Raman software to start a scan and then traverse the scan through the Raman neural network. The main

program freezes for the duration of the scan and longer if the function is called without the use of threading.

3.4 Camera Setup

Each camera that has been tested has its own software development kit that allows the user to interface with the camera and adjust parameters. Multiple parameters need to be set to get the optimal performance for the current system.

The initial variables that needed to be set were the width/height and pixel format for the images. The width/height was set based off of the viewing tube size within the frame. The width was set to 2000 and the height was 875 to cover the full height and width of the inside channel without the bottom and top barriers of the tube showing. The pixel format was set to BayerRG8 to allow the fastest streaming in color. The stream channel packet size, stream packet resend, and acquisition frame rate were enabled. The GevSCPS packet size was set to 16000 which dictated a higher bandwidth and allowed the program to not crash by attempting to send too much data at a time for the requested frame. The buffer handling mode for streaming was set to newest only. The auto exposure was set to continuous. The trigger overlap was set to the previous frame. The auto gain was set to continuous, and the gamma was set to 1.5 for the optimal display of the image. The device power and temperature were displayed before the main loop and then the stream of images was started with an initial 100 free buffers. Each image is then grabbed from the camera using a GigE connection for the duration of the program with a set timeout.

3.5 Object Detection

Once the frame was collected from the camera, the image needed to be processed to detect any objects within. Different methodologies have been used for this process such as

incorporating traditional approaches like background subtraction with contour tracing and the support vector machine, but these are computationally expensive and the accuracy of differentiating types of objects are less than the modern approach of detection which is using deep learning. The benefit of using traditional approaches or unsupervised learning methods, is the user does not need to collect a database of the categories to detect an object. This project needs to detect each object and determine the location in real time to initiate a Raman scan efficiently. A fast process of detection is needed as well as accurately defining what type of particle is passing through.

This project is meant for field use and particles such as dirt or algae can still enter the system. The ability to differentiate between microplastics and dirt is beneficial to not have large amounts of scan data on incorrect objects. The morphology and size of these microplastics is also valuable information. The morphologies of the microplastics that were focused on were the fiber, fragment, and pellet morphologies. Dirt particles were primary objects that could pass through the system. They have varying morphologies and aren't the focus of this project so they were bundled into one category labelled as dirt. Algae also can pass through the system, more so dependent on the testing sites, and have many variations of morphologies. There was a difficulty in collecting and classifying each type of algae in the current system, therefore this category was left out. The main categories that were needed to be detected in the system were the fragment, pellet, and fiber morphologies of microplastics as well as dirt to set a total of four classes.

The (YOLO) detection system was the primary neural network used in this project. Varying models have been trained in YOLOv4 and YOLOv5. Resnet 10 and Deceptron instance segmentation models have been created as well to compare and to use the instance

segmentation to gather more accurate sizing information that the bounding box lacks. The traditional approaches, neural networks, and the methodologies of data collection will be explained in the next subsections.

3.5.1 Traditional Object Detection

The background subtraction method was the first computer vision technique used in testing before transitioning to deep learning. It uses an initial frame as the reference image to create the background that does not change throughout the test, every change within the frame afterwards of a set threshold will be isolated as foreground objects and compared with the background image to obtain the difference and to create a distance matrix. The background in this project was considered as the viewing tube and any foreground objects were the categories of objects needed to be detected. A threshold value was used depending on the lighting within the frame from the LED. A contour can then be created which is a connected set of pixels surrounding the foreground object. Figure 3.3 represents the process for background subtraction and an initial image using the technique with LabView and the grayscale camera. The object within frame is a PE pellet which is used for most of the testing.

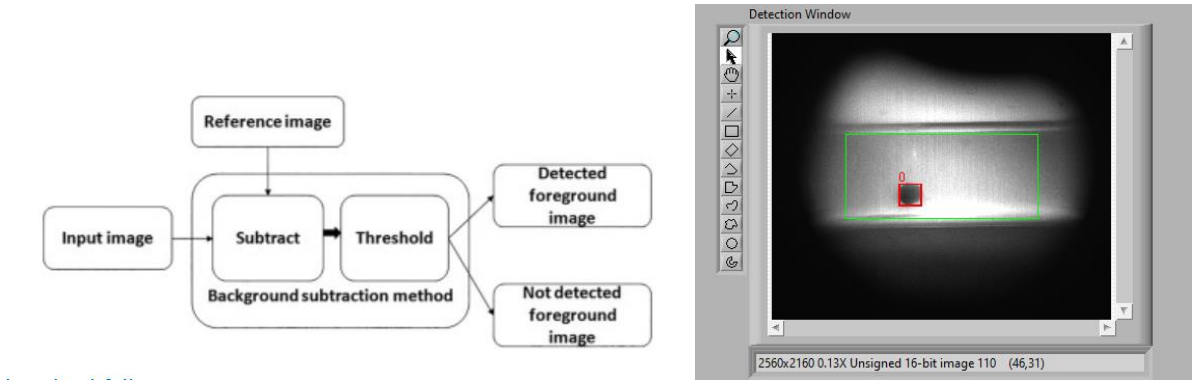


Figure 3.3 Background Subtraction Implementation

3.5.2 YOLO

Since this project needs a real-time response with better accuracy in differentiating between morphologies of the objects, the YOLO architecture was chosen. The workflow for the object detection system from data collection to the results of the output is shown in Figure 3.4.

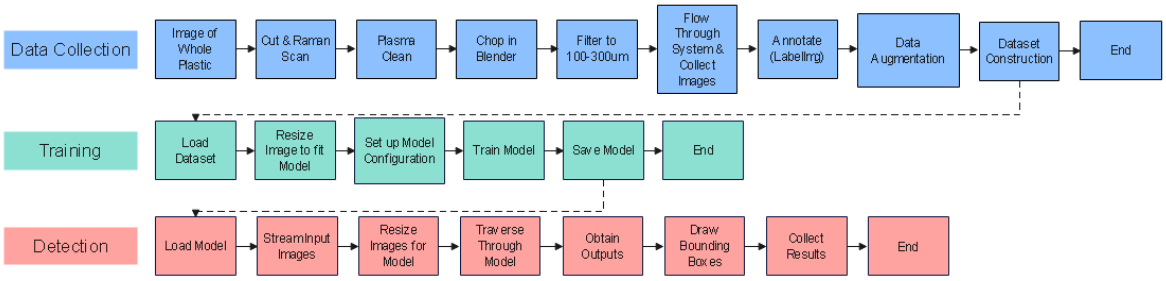


Figure 3.4 Object Detection Process

3.5.6 Data Collection

Each category that needs to be analyzed need to first be collected and imaged. The fragment morphology of microplastics was obtained by collecting household products from three different plastic types that were easy to obtain and image: PET, HDPE, and PP. LDPE is primarily used in bags and PS as Styrofoam. Bags and Styrofoam objects were difficult to cut down into micro particles, therefore they are excluded. There has been a lack of PVC products that have been accumulated, therefore it was excluded as well. There is a large number of plastic types within the Other category, therefore there is not a focus in this category for now. The plastics that were obtained were first imaged as a whole. A few plastics that were used for imaging are represented in Figure 3.5.

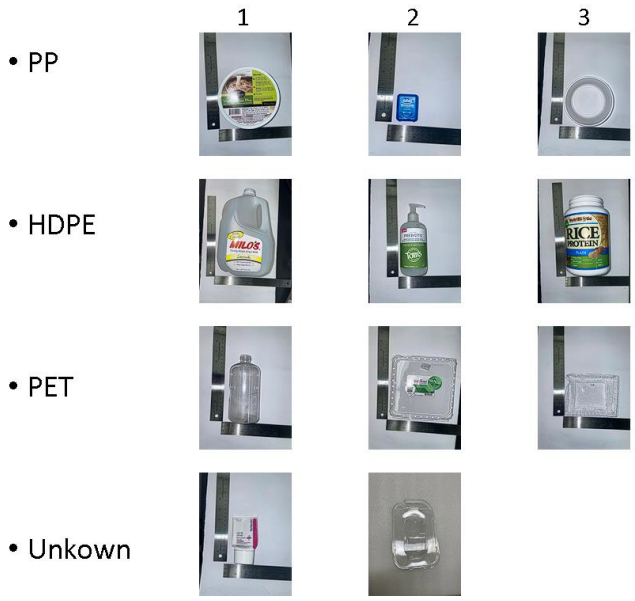


Figure 3.5 Plastic Samples

Once the plastic was imaged fully, a piece was cut out and scanned using the spectrometer to collect the scan data and to make sure the classification was the same as what was represented. This piece was labelled and then logged within the Raman database. After the plastics have been fully imaged and scanned, multiple small pieces were cut out and then placed inside of the Plasma Cleaner shown in Figure 3.6.

Figure 3.6 Plasma Cleaner

It is better for the plastics to be more hydrophilic to suspend in the water and flow them into the system without them getting caught within the hosing. Plasma coating was a

great way to make the surface of the plastics hydrophilic. Hydrogen, oxygen, argon, nitrogen, and ammonia gases can be used to treat the surface to clean, make hydrophilic, and improve the bonding strength. Work done by Lai, Jiangnan, et al. [2] shows the relationship between the amount of plasma treatment work with the contact angle of PET, PP, and PC that results in the three polymers' hydrophilicity increasing per amount of work. The contact angle reduces as work is increased which indicates a higher hydrophilicity.

Once the pieces were plasma cleaned, they were then chopped and grinded down into the micro range by using a blender and two mesh filters to obtain the range of 100-300µm. This process of chopping and filtering continued until a large amount of microplastics were visible within the collection beaker. An example of the chopped microplastics within the filter can be seen in Figure 3.7.

Figure 3.7 Filtered Microplastics

The pellets that were used in the dataset are three sets of Cospheric branded industrial Polyethylene beads. The first set was a green color in the range of 106-125µm at 0.98g/cc, the second set was a black color in the range of 212-250µm at 1.2g/cc, and the third set was a red color in the range of 250-300µm at 0.98g/cc. The fibers that were used were primarily fabric material found from clothing in laundry. The dirt used within the

dataset were samples from the Warrior River that were collected and then filtered out inside of the lab.

The filtered samples were then pumped into the system to image. Within the main program, there is a mouseCallBack function that allows the user to save the current frame by left clicking the image window. The file pathing name of the category was incremented after the frame was saved based on the number of images that have been collected i.e., fragment1, fragment2, fragment3. Examples of images for each category is represented in Figure 3.8. Examples of the images collected in grayscale are represented in the Appendix.

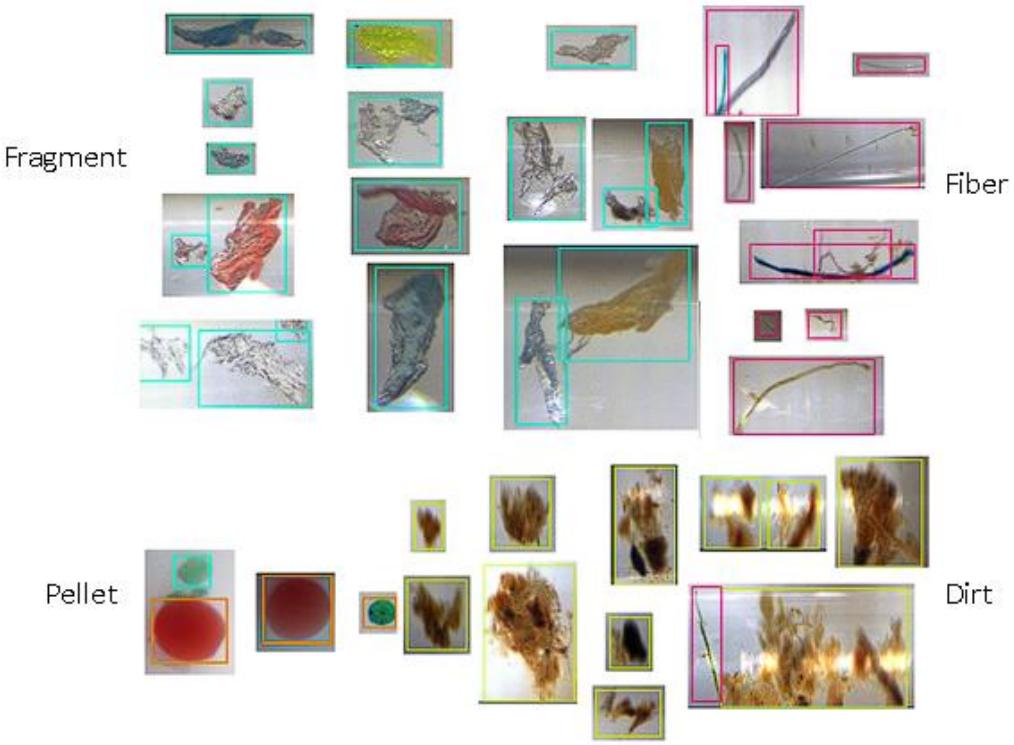


Figure 3.8 Colored Dataset Examples

Once enough images were collected for each category to create a testable model, these images were then annotated using the Python application LabelImg. A bounding box was drawn around each object for the main models and a multiline bounding area was drawn

for the instance segmentation model. The category label along with its point coordinates was saved into an xml file for each bounding object. After each image has been annotated, more images can be generated from the current library through augmentation in Python as well as through the easy-to-use website Roboflow. Augmentations such as an added exposure of -15% and +15%, a random gaussian noise added of up to 1% of pixels, a 1.5% percent blur, and an added contrast of -15% and +15% have been used to generate more images. A table displaying the total number of color images collected for each category and total amount after augmentations is displayed in Table 3.1. The table of grayscale images are in the appendix.

Color Images

Class	Fragment	Pellet	Fiber	Dirt	Total	Augmented Total
Images	5397	131	2571	680	6886	16457
Annotations	9958	243	3498	1122	14821	

Table 3.1 Color Image Distribution

3.5.7 Training

Training was done through Google Collab. Each image was first resized to the parameters needed for training which was at 416x128 for the model used. The batch size of 128 is set and training lasted for 232 epochs which gives a total 79.3 mAP50 rating shown in chapter 4.

3.5.8 Preprocessing & Postprocessing

After a model has been trained and saved, the weights can be used for inference. The model was loaded through OpenCV's dnn library. The image was converted to an OpenCV

mat variable, was resized to fit the model at 416x128, and then the image was used as input for the model. The output vector of the mat variables was postprocessed. Only the maximum confidence scores located within the output were saved. Non maximum suppression was used to eliminate redundant overlapping boxes with lower confidences. The box's class label that had the highest score was assigned. The bounding box can then be drawn using the box coordinates of the detected object. The variables that were needed to be saved within postprocessing were the count, timing of detection, label, height, width, and confidence. The GUI was updated with the counts of each class as well as the confidence score of the detections.

3.6 Raman Scanning

The Raman spectrometer was activated through the Symbion software. To operate the spectrometer through the main C++ program, a component object model (COM) initialization needed to be created with the use of combaseapi.h for the Symbion application. This allows the user to create binary software components that can interact with each other. COM is the foundation to ActiveX and OLE technologies. The ID of the application is obtained and then the control functions can be accessed after an instance is created. The variables needed to be submitted to the function were the driver title of the spectrometer, scan sequence to occur, exposure time of the scan, and the save path of the spectrum.

The bounding boxes that were drawn around each object have the pixel location of the top, bottom, right, and left sides of the box. The sample was flowing through from the left to right side of the tube so the first point of contact to the laser with the object is its left side. The laser needs to be targeted on the object for most of the duration of the scan to get an accurate result that can be classified. The location that received the best results is the

range of 200 pixels before the scan location which is 350 µm as the pixel size is 1.75 µm in size. This gave enough time for the scan to be initiated and fully scan the object with a flow rate of 0.061 ml/min and an exposure time of 0.3s. Once the object was within this range and the label was not dirt or fiber, the LED is turned off and the thread for the Raman scan was initiated. The spectrum file that was saved through Symbion was sent through the neural network and the classification of the scan was determined. The GUI was then updated. The LED gets turned back on once the scan has finished and the full process repeats until the time has been executed.

3.6.1 Raman Neural Network

A 1D neural network for the Raman data was created through Python using TensorFlow's Keras API. It used a sequential model with ten layers. An overview representation of the model is displayed in Figure 3.9.

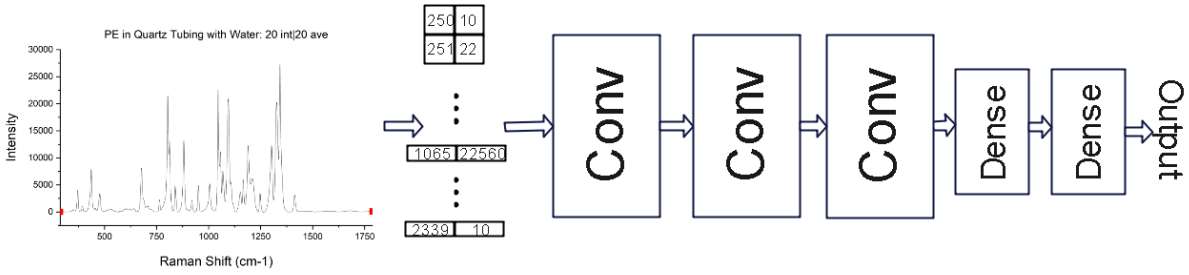


Figure 3.9 Raman Neural Network

There are three convolutional layers with three following max pooling layers. Each convolutional layer used the Relu activation function. After the sequence of convolutional layers were done, a flatten layer was set to flatten the input. A dense layer followed with a set parameter of 500. A dropout layer was used to account for overfitting with a dropout rate of 0.5. A dense layer was then used with the SoftMax activation function to collect the final

output for the number of labels desired. The Raman spectra was converted into a text file with just the x and y data points. The y data points represent the peaks within the spectra whereas the x data points represent the Raman shift from 250 to 2088. Each file for training was concatenated into a single data matrix. Once the data from the divided categorized folders are collected, the model is then compiled.

3.6.2 Data Collection

The process of collecting Raman data is similar to collecting the microplastic images, but without having to plasma clean and chop the plastics into micro particles. The whole plastic was first imaged with a white background and ruler. A smaller piece of the plastic was cut and then scanned through the scanning software Symbion or EZ Raman. Another piece was cut out if there were varying parts of the plastic such as a cap to a container with a different color. An efficient integration and average time of three seconds was used for each scan and three different positions of the plastic piece were scanned for a total of three scan files per plastic cut. The plastic piece is then labelled and stored within the storage container that the user has labelled for their data. Table 3.2 shows the total number of scans used for training the neural network.

Raman Trained Data

1:PET	2:HDPE	3:PVC	4:LDPE	5:PP	6:PS
160	98	9	61	121	18

Table 3.2 Raman Trained Data

PVC and PS are the undervalued plastics within the dataset as these are the least common products brought into the lab. Thin LDPE bags do not always give an accurate scan and sometimes need to be cut at different locations. The Other category is left out because of

the large number of varying plastics that are within this category have different peaks that are difficult to classify. Examples of scans collected using the exact plastic samples in Figure 3.5 can be seen in Figure 3.10.

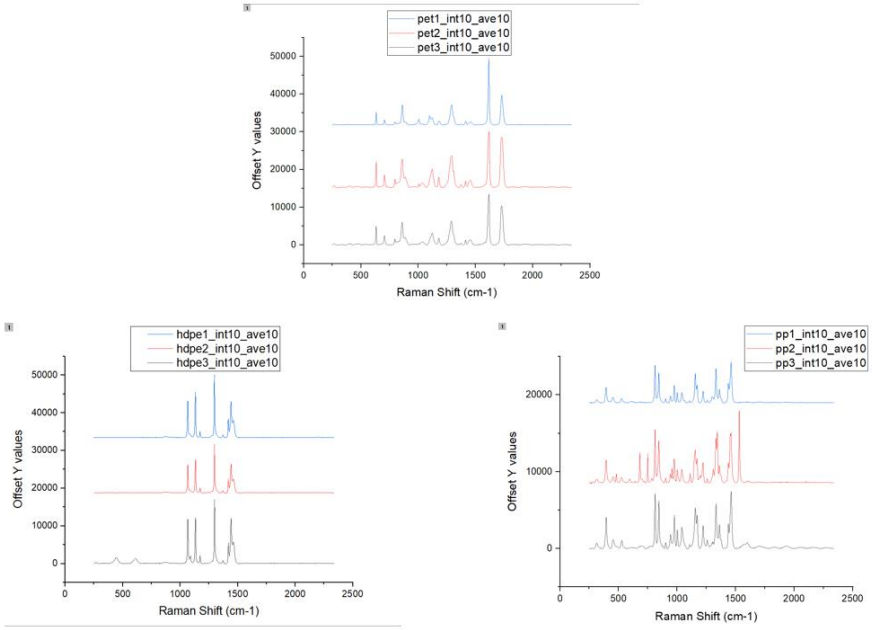


Figure 3.10 Raman Spectra of Plastic Samples

An integration and average time of ten seconds was set for these scans, and each give accurate scans for the recycle code that was given.

3.7 Data Management

In the current iteration, all the output data was saved to the local machine or an external hard drive. A csv file with the detected object statistics that include the time, count, and size were saved. The output video was saved for checking the accuracy of the detections and to further analyze the types of particles that traverse through the system.

References

[1] Cardoso, Francisco. *An ImGui for a Cooperating Vehicle Simulation framework*. Diss. ISEP, 2019.

[2] Lai, Jiangnan, et al. "Study on hydrophilicity of polymer surfaces improved by plasma treatment." *Applied Surface Science* 252.10 (2006): 3375-3379.

CHAPTER 4 RESULTS AND DISCUSSION

4.1 Overview

In this entry portable system, the focus in total results is based on the neural network accuracy in detection of objects and the accuracy of classifying the Raman spectrums, the synchronization of scan timing, and the field test findings. Results in filtering efficiency and improvements in the timing of Raman scans as well as color detections will be the initial focus in the next iterations of this project. The responsibility of gathering filtering efficiency results has been designated to other students who have helped with the project; therefore this is left out within this thesis since the work was done by others. The progression of the portable system in field testing will then be improved upon once the consistent scanning efficiency is made with the detections.

4.2 Object Detection

The results of training a neural network highly depend on the size of the dataset, annotation quality, complexity of the objects within the images, and the hyperparameters that are used for training [1]. Training was done using Python and PyTorch within Google Colab. Colab's GPU resources and CUDA were incorporated to speed up the training process. As the grayscale dataset was previously used in this research, most of the count and flow tests were done with the grayscale models. A YOLOv4 and deceptron model was created for the grayscale model as well as the use of contour tracing and background subtraction. The YOLOv4 model gave an accuracy of 81% and the deceptron mode showed

80%. The YOLOv4 model is used for the count results and more testing with the color models need to be done. A comparison will be made with the YOLOv5 and detectron instance segmentation results for the colored dataset along with the count results with the YOLOv4 model in grayscale.

4.2.1 YOLOv5

The image size for training was defined as 416. The batch size is one of the most impactful parameters to set for neural networks. This parameter represents the number of samples that propagate through the network and is crucial when fitting the model. The smaller the batch size is, the faster computation is, but it needs more images to reach the level of accuracy as there are less additions per training iteration [2]. A batch size of 128 was chosen for the model represented here. Additional models of differing batch sizes and image sizes were made with little to no difference in the training results. The YOLOv4 training results are displayed in Figure 4.1.

```
Model summary: 157 layers, 7020913 parameters, 0 gradients, 15.8 GFLOPs
               Class     Images  Instances          P          R      mAP50   mAP50-95: 100% 11/11 [00:08<00:00,  1.25it/s]
                 all       1316       2866      0.853      0.726      0.794      0.519
                dirt       1316        222      0.865      0.716      0.793      0.501
               fiber       1316        611       0.79       0.44      0.552      0.274
            fragment       1316       1983      0.852      0.768      0.842       0.54
              pellet       1316         50      0.904       0.98      0.989       0.76
```

Figure. 4.1 YOLOv5 Results

The model was tested with a total of 1316 images and a total of 2866 object instances. The division of how many instances per class is shown in Figure 4.1 along with the precision (P), recall (R), and mean average precision (mAP). Precision attempts to answer how many positive identifications were truly correct. Recall attempts to answer how many true positives were identified correctly. The mAP shows how well the model detects objects at different levels of confidence. The number after mAP represents the IoU thresholds used. The mAP50 uses an IoU of 0.5 whereas the mAP50-95 uses a range from

0.5-0.95. These are useful metrics to track the performance of the model. The fiber class had the lowest accuracy among the four classes. A more diverse representation of fiber images are needed to help improve this, as the majority of the colored fiber images that have been collected so far are embedded with fragments surrounding it or the images are the grayscale fibers. This model achieved a fps of forty-two without the use of a PCI adapter, and a fps of seventy with the PCI adapter. Examples of detections using the YOLOv5 model can be seen in Figure 4.2.

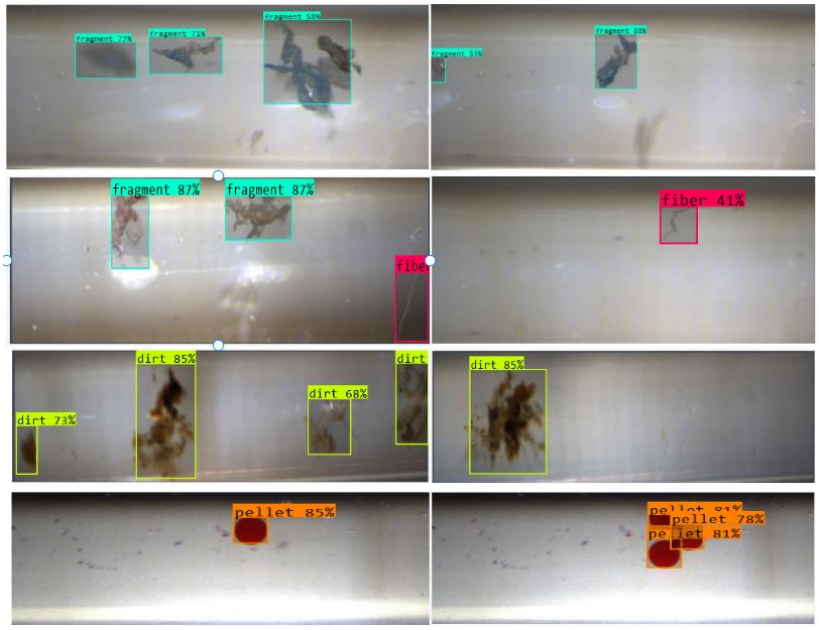


Figure 4.2 YOLOv5 Detection Examples

4.2.3 Instance Segmentation

The Detectron RCNN instance segmentation model was used to give a better representation of the size for the object with the outline encompassing the whole object as compared to a bounding box which is not as suitable for complex oriented objects such as fibers. The instance segmentation model does not run as fast compared to the YOLO model, with a fps of ten. It is not suitable for real time testing; however, the model can still be

incorporated within the program to test on the saved output video and compare size and detection results with the real time model that is used. This model's accuracy was much poorer compared to the YOLO model, with a 53% mAP. This low accuracy could be due to the fact that the images were annotated with student help, and not all images were annotated or annotated well. Figure 4.3 displays test images through the instance segmentation model.

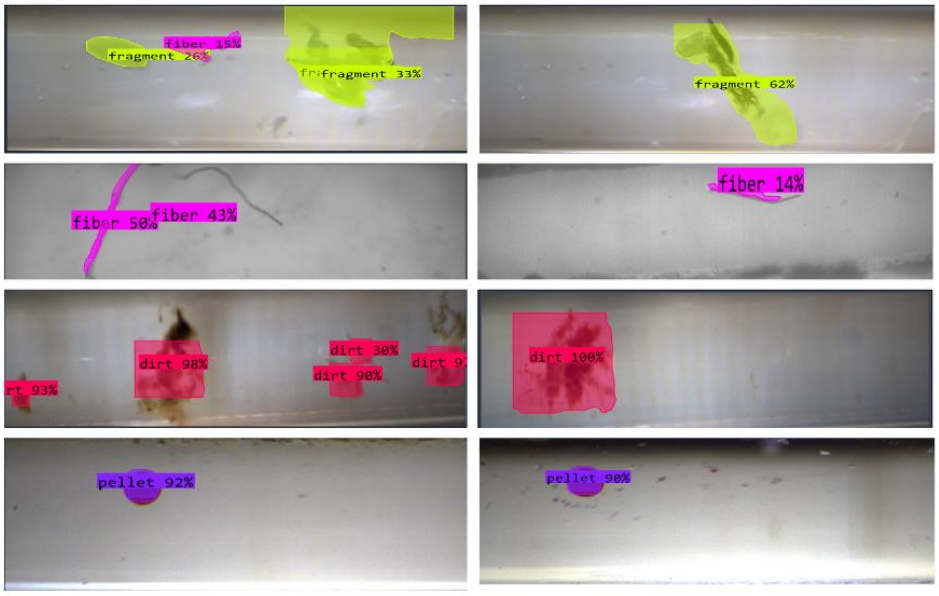


Figure 4.3 Instance Segmentation Examples

4.2.4 Counting

The counting accuracy was determined by flowing a set number of particles through the channel at different flow rates. The contour tracing background subtraction method was used initially to test the count and then the YOLOv4 model was incorporated with count testing. A PE pellet was passed through the channel. The flow rate was adjusted to test the limit speed of where the model started to give error in count efficiency. Another difference from the grayscale Zyla camera and the color Lucid camera was the FPS they could achieve with the same sized frame. The grayscale model with the Zyla camera achieved a fps of up

to sixteen, whereas the color YOLOv5 model achieved an fps of forty-two without the PCI adapter. This could lead to different results and flow rates for the two systems. The count was tested at flow rates of 0.1, 0.2, 0.4, 0.7, 0.8, 1.0, and 2.0 ml/min with a mixture of PE pellets and fibers. The count efficiency was 100% for flow rates 0.1-0.8, but missed three out of twenty-five objects at a flow rate of 1.0 ml/min and missed 90% of objects at 2.0 ml/min. As the speed increases past 1.0 ml/min, the objects become distorted passing through the tube. The model's fps of around sixteen could allow the object to pass through the frame window and it not be detected by the camera when the flow rate becomes too fast. An example of the results output which include the total count, labels, size, and time of detection with the flow rate of 1.0 ml/min is seen in Figure 4.4.

```
Flow Rate: 1.0 ml/min
Exposure Time = 0.001;
Window Size = 2560x600
 Acquired Frame Rate = 250

                Total Object Count: 22
------------------------------------------------------------
    ID            Width       Height         Time (s)

1)  fiber          274         366            7
2)  fiber          99          107            17
3)  fiber          304         203            17
4)  bead           225         331            18
5)  fiber          452         69             20
6)  fiber          197         216            21
7)  bead           263         266            21
8)  fiber          328         95             22
9)  bead           275         268            23
10)  fiber          371         255            24
11)  fiber          310         242            24
12)  bead           244         249            25
13)  fiber          133         152            25
14)  fiber          433         440            26
15)  bead           108         263            28
16)  bead           223         246            29
17)  bead           260         264            31
18)  fiber          215         347            31
19)  bead           253         292            34
20)  fiber          124         165            49
21)  bead           261         244            50
22)  bead           229         303            56
```

Figure 4.4 Output of Detections using a 1.0 ml/min Flow Rate

The counting method used the bounding box output array and the difference of the number of objects within the current frame from the previous frame as the indication to increment. This means that with less accurate models that continuously apply a bounding box to the same object off and on within frames, the count will increment more than needed

and count the same object more than once. This counting method is not the most efficient when dealing with inaccurate models. Count methods outside of this would dictate further processing that can slow the frame rate down. This method was also inefficient for Raman scanning, as after the scan finishes, the bounding box is reapplied to the object which increments the count. This is an area that can be improved upon in the future with the comparison of speed with accuracy in different counting algorithms in neural networks.

4.3 Raman Scanning

The efficiency of the Raman neural network and the scan consistency with plastics flowing through the tube will be observed. Close to 10% of the 461 samples within each class that are shown within Table 3.2 were used as test samples and 90% were used for training for a total of 47 and 461 samples respectively. Training with Python and TensorFlow using the adam optimizer, categorical crossentropy loss function, 100 epochs, and a batch size of 50 translated to a 90% accuracy for the model. Out of the 47 test samples, two of the samples were classified incorrectly. The two samples were labelled as a four (LDPE) and two (HDPE). These were classified as a five (PP). After further examination of the two spectrums, they were labelled correctly, however some of the scans within the library were not formatted the same as can be seen in Figure 4.5. This displays the two spectrums in question with a reference PP scan. The intensity levels of the LDPE scan are much lower, and the baseline of both scans were not corrected to zero such as the PP scan, which could lead to inaccurate classifications. This gives an insight on the Raman data that has been collected within this library. It has been a group effort and there can be differences with how the scans were initiated and saved.

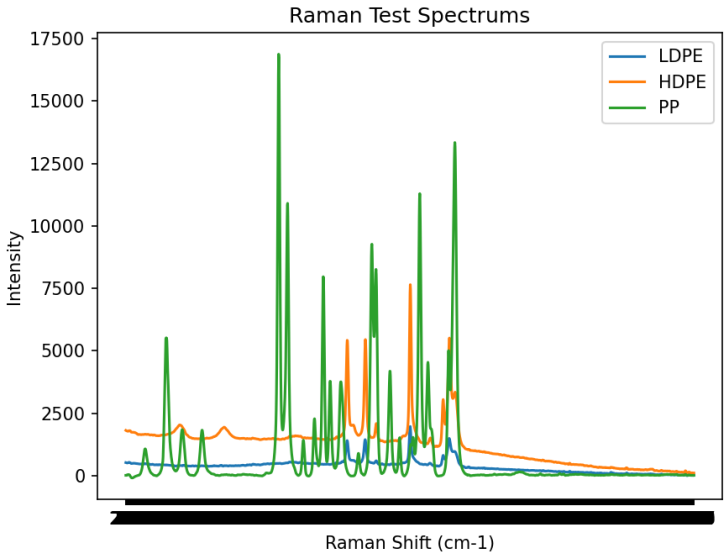


Figure 4.5 Raman Test Comparisons

A first test was done to show the efficiency of scanning at different flow rates with the spectrometer continuously scanning with a 0.1 second integration time, which is the fastest scan time that can get a reasonable spectrum to be identified. A scan time of 0.3 seconds was usually used within the program because it had higher intensity spectrums that can be classified much easier. PE pellets traversed through the tube for a total of fifteen times with the flow rate incrementing from setting one to five of the mini pump that is used within the cart. The plastics become much more difficult to scan past setting five because of the speed of the flow rate and initiation time of the scan. Flow rates of each setting starting from one to five as well as the fifty and ninety-nine control settings can be seen in Table 4.1. The fifty and ninety-nine settings are included because these settings are sometimes used to speed up the flow rate while testing and collecting pictures for the dataset. Each setting was tested for thirty minutes.

Setting	1	2	3	4	5	6	50	99
Rate(µl/min)	57.4	60.9	64.3	67.5	72.1	73.8	486.2	615.4

Table 4.1 Mini Pump Flow Rates

A collection of scans was obtained for the first five flow rates, these scans were parsed within Origin. The scans with detectable peaks were the scans that successfully hit the PE pellet. These scans are displayed in Figure 4.6. The titles detail the incrementation through the five flow rates and how many hits of the pellets were obtained.

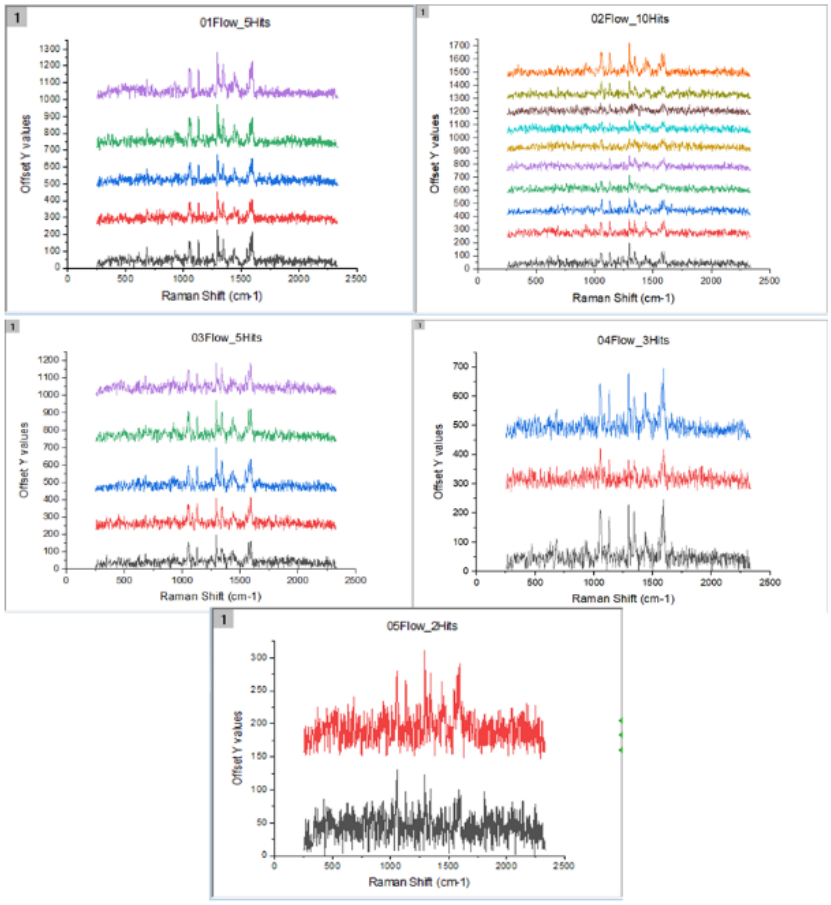


Figure 4.6 Raman Scan Flow Test

The most hits were with setting two of the mini pump. The bottom black scan for setting five and the red scan within setting four barely hit the PE based on the poor signal. The red scan within setting four was successfully classified with HDPE, however the black scan was classified as PP. At a flow rate of 60.9 µl/min which is 1.005 mm/sec, the pellet passes through the total viewing area for twelve seconds. The viewing area is a total of 2000 pixels in width and a pixel is the equivalent to 1.75 µm with the current positioning of the camera. The total width of the viewing tube is 3.5mm which gives a flow rate of 4.861 µm/sec or 1.489 µl/min for a 275 µm PE pellet.

The second test used the standard method of scanning in the system which uses the bounding box location to initiate a scan. The range of initiation is 200 pixels or 350 µm. The range ending is in the middle of the scan position and the start of the range is on the entry side of the tube with the microparticles flowing from left to right. The first four flow settings were used in this test as the fifth flow setting had inferior results in the first test. In a continuous run, four pellets were traversed through the tube starting at the fourth setting down to the first setting for each pellet. Each pellet is separated a few seconds apart from each other so that only one pellet is in frame at a time for each flow rate. The scans for flow rates three and four miss the pellet. The scan hit the pellet in settings one and two, however the scan is initiated twice in setting one. The test is reproduced but with each flow rate being tested with five pellets instead of one. Again, scans for settings three and four miss the five pellets, but the scans hit each time for settings one and two. Setting two scanned twice for one of the five pellets and setting one scanned twice for every pellet that passed.

A proper test still needs to be made with multiple pellets close together flowing through. From general testing, particles close together give issues with the scan efficiency. The scan

integration time is regularly set to 0.3s, however this time is not accurate in the length of time for the scan. The time of a total scan is measured to be 1.5 seconds in the Symbion software and 2.2 seconds when initiated through the program which gives issues when starting scans for pellets that are within twenty micrometers of each other. When many microparticles are aggregated together, they can get displaced throughout the channel vertically which does not align with the center Raman laser positioning. The scan timing has not been perfected in this system and still needs to be improved.

4.4 Field Testing

Field tests have been conducted primarily in Tuscaloosa, Alabama at the Warrior River and Lake Lurleen. The Warrior River was tested for a total of thirteen hours, while Lake Lurleen was tested for a total of five hours. In addition, eight hours of testing was conducted in Michigan. The goals of the field tests were to determine the efficiency of the system in detecting and classifying microplastics out in the field from river water and to test its general portability and operation. The possibility of no microplastics being detected was accounted for and more locations need to be tested such as oceans and wastewater treatment plants.

During field tests, most of the particles that passed through the system were smaller than what was tested in a lab setting and what the object detection models were trained on, which resulted in an inaccurate measurement of the total number of particles that passed through. Upon further analysis of the saved videos from the tests, not many microplastics within the tested range were found. Since the object detection program was not designed to recognize the small particles that passed through the system, the videos had to be manually reviewed in order to count each particle. It was challenging to accurately label the small

particles with the naked eye because of the magnification set, so all objects passing through were labeled as "particles" and the size distribution was tallied. Particle representation for the Warrior River can be seen in Figure 4.7. Particle representations of testing done at lake Lurleen and in Michigan are shown in the Appendix. These particles that were large enough to be detected, were primarily labelled as dirt or fragments. Upon further inspection into the fragment detections, some of the detections were red fragments such as chipped pellets that were lodged within the system or possible particles from the red cyclone. Most were misclassified dirt particles. Further inspection is needed into the final output from field testing.

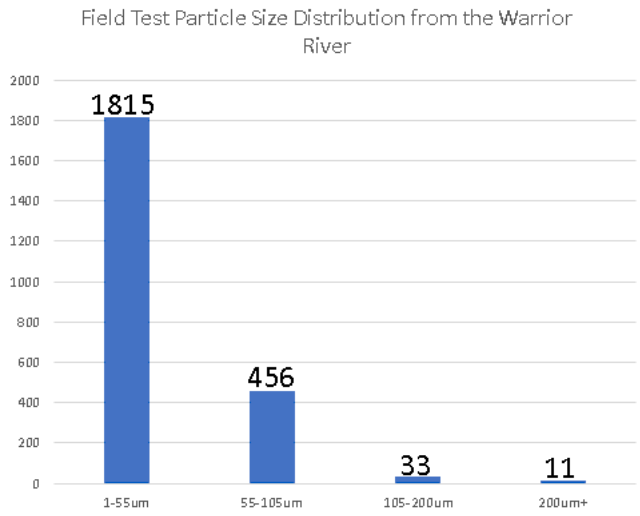


Figure 4.7 Warrior River Particle Representation

Field testing was not successful in terms of not examining any microplastics within the range that has been focused on in this project, but this is a good thing as one would hope to not find many microplastics in our local river. The project does need to be adjusted to test with a smaller scale of microplastics if it is progressed for testing the local water in Tuscaloosa.

4.4 Discussion

The results of this thesis show the accuracy of the neural networks to detect objects and to classify microplastics. The YOLOv5 model achieved a mAP of 79% and the instance segmentation achieved a mAP of 53%. The models were especially inaccurate when attempting to detect fibers. More diverse images of fibers are needed and the images that have been annotated thus far need to be double checked to make sure each fiber is annotated sufficiently. Fibers are expected to have the lowest accuracy because of their many different configurations that can appear within the channel, the contour tracing and background subtraction methods also could not correctly detect the whole fibers on a consistent basis compared to whole fragments or pellets. The YOLO models have a fps that can detect objects in real time and can efficiently initiate Raman scans within the time range of the passing microplastic. With a fps of around ten, the instance segmentation model would not be sufficient to detect objects in real time for Raman scan initiation, but there is still interest of using the network after the end of the program to gain a better understanding of size information for the objects that are detected. An importance of this arises especially for fibers since they do not cover the majority of bounding boxes.

The count results start to become less efficient after 1.0 ml/min with the previous YOLOv4 model which does not become a problem as the flow rate to collect an efficient Raman scan is within 57.4 to 72.1 μl/min. The flow rate is never increased more than this with the mini pump unless to researcher is not using the Raman scan functionality. It is increased to speed up the process of collecting images for the dataset and general testing to push the microparticle into frame quicker to test the object detection or to initiate Raman scans of objects within the tube. It is also used to dislodge microparticles that can get caught

within the tubing or to speed up the very beginning of field tests before there is water flowing through the quartz tube.

The Raman scanning neural network achieved a high accuracy, but the signals that have been collected are those that do not have much noise, and the noisy spectrums would be beneficial to try and differentiate. The main issue with the neural network is the lack of total data collected. The scanning initiation timing and efficiency can be improved on by stabilizing microparticles within the tube for a consistent scan. The timing of the Raman scan to gain a valid clear spectrum does limit the flow rate by a large margin for a continuous flow. This project does remove the physical work that the user has to apply to filter, visually identify each microplastic, and physically navigate the scan to the correct position.

References

[1] W. Zhiqiang and L. Jun, "A review of object detection based on convolutional neural network," *2017 36th Chinese Control Conference (CCC)*, Dalian, China, 2017, pp. 11104-11109

[2] Schmeiser, Bruce. "Batch size effects in the analysis of simulation output." *Operations Research* 30.3 (1982): 556-568.

CHAPTER 5 FUTURE IMPROVEMENTS & CONCLUSION

5.1 Size

The system is efficient in a controlled setting for microplastics within the range of 100-500µm, however the particles that mainly pass through the system during field tests are less than 100µm. The total size of the particles that are focused on can be reduced with smaller tubing, more efficient Raman scanning positioning, and focusing of the microscope for smaller particles that are flowing through.

The current size of the deployed system can be made smaller by investing into devices with reduced size such as the spectrometer, computer, and power supply. The cart is made of steel which makes it heavy and can be replaced with a lighter material as well.

5.2 Input

The current input to the whole system only intakes right below the surface of the water near the shoreline with a small input size of 3/8” diameter. This does not cover a lot of area at the testing site and can be improved by using a multi-port setup or a manta net. This multi-port setup can be designed to cover the horizontal and/or vertical depths within the testing site to allow for a larger area to be sampled as well as testing to analyze the samples at different depths. One benefit of sampling microplastics at different depths is that it allows for a more comprehensive understanding of the distribution of these particles in the water column. Microplastics can be found at various depths in the water, from the surface to the bottom of the ocean floor. The manta net allows for a large area to be sampled at the surface

of the water as well and includes the mesh filtration design to contain particles which can be cleaned and analyzed after testing. The input samples can also be improved by testing at different locations such as oceans and wastewater treatment plants as well as different weather conditions.

5.2.1 Fluid Flow & Particle Control

Improvements on the positioning of microplastics within the quartz tube can be made by adding microfluidic chips, multi-input ports, valves, or flow meters to control the fluid flow. Using DEP that applies electrostatic fields to attract or repel particles in water to navigate to the scanning position. A new control pump can be invested into or designed to allow a more consistent flow through the tube as well as stopping,, incrementing /decrementing the speed, and reversing the fluid flow using the program to control it remotely instead of manually. A swivel designed under the Raman spectrometer that is controlled to move according to the detection of the microplastics could possibly provide more efficient scanning of each object that passes instead of having a single spot that needs to be targeted. The flow rate could possibly be improved by using a high flow rate that pauses to scan on an object detection and then continues afterwards or by using a more efficient scanning software or protocol that can gather data quicker. The overall control of the particles flowing through the viewing tube is essential to have an optimized system for detection and classification of microplastics using imaging and Raman spectroscopy.

5.3 Programming

The object detection and Raman classification accuracy can be improved by adding more data to the training set. Specifically, fibers for the object detection and plastic types 3 and 6 for the Raman classification. Once more data is added and the accuracy is improved,

more classes can be included to the detection models such as the microplastic morphology type foam and variations of algae as well as the plastic type 7 for the Raman classification model.

The tracking and count of objects can be improved with a more efficient model or by using a tracker algorithm like DeepSORT, ByteTrack, or BoT-SORT with the possible cost of a frame rate drop [1, 2, 3]. DeepSORT has been initially tested with and one of its key features is its ability to handle objects that block the view of other objects. It also performs well with multiple objects within the frame which is beneficial for this system as multiple objects can pass at a given time.

The GUI can be improved to include the image frame window instead of it being separate from the statistic display, library of images collected that can be viewed individually, and Raman classification accuracies and spectrums instead of just the label and count.

5.4 Data Management

The data management can be improved in this system by adding a cloud-based synchronization of data. The output data of the system can fill the memory of the local devices and having more sources to unload the data is optimal. Using the cloud allows the research team to visualize the data off site and monitor it together. The data is also more secure compared to a local machine that can experience data loss due to device failure or accidental deletion as well as including security measures to protect against data breaches or unauthorized access.

5.5 Conclusion

The work done within this thesis lays down a foundation that can be improved upon to develop an efficient portable system that sorts, detects, and classifies microplastics. The research in the analysis of microplastics is a worthwhile venture as studies start to mount up on the accumulation throughout the world and a there is a demand for easy-to-use automated systems to do so. The current results of the limited field tests were not the best and there is a lot of room for improvement in the current system, but this research can potentially lead to an automated system that will remove busy work out of the equation for analyzing microplastics in the future.

There is a vision that the researcher turns the power on and connects the input of the device into any sample and can leave the rest of the work to the system as it sorts and gathers information on the microplastics with a graphical user interface displaying the statistics of the objects detected such as the total count, size, and morphology as well as the classifications made by the Raman spectrometer. The current system can achieve this, it just needs to be improved for consistency.

References

[1] Narinder Singh Punn, Sanjay Kumar Sonbhadra, Sonali Agarwal and Gaurav Rai, Monitoring COVID-19 social distancing with person detection and tracking via fine-tuned , Computer Vision – ECCV 2022, pp 1-21

[3] Nir Aharon, Roy Orfaig, Ben-Zion Bobrovsky, BoT-SORT: Robust Associations Multi-Pedestrian Tracking, 2022

APPENDIX

This appendix includes figures and tables that represent some of the work done in this project that is not included throughout the main chapters to give a better idea of the progression and materials used for research.

Figure A.1 Starting Point with a Pump, Zyla Camera, and Microscope

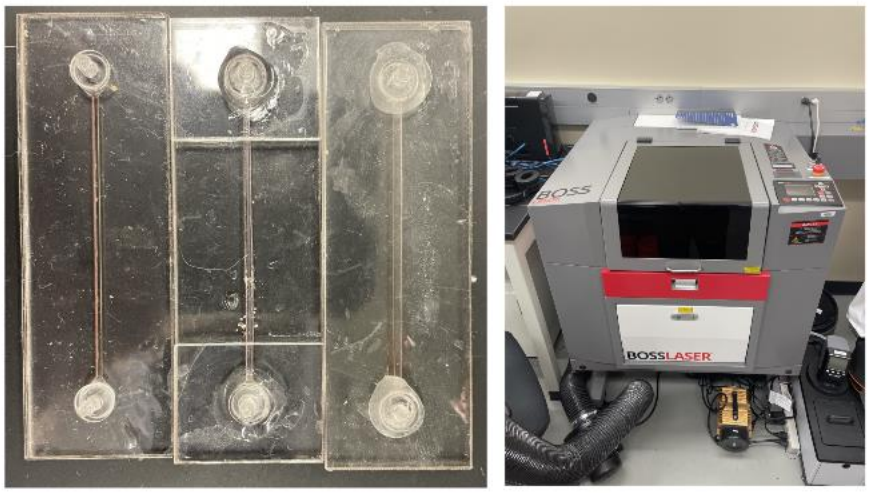


Figure A.2 Starting Channels and Laser Cutter

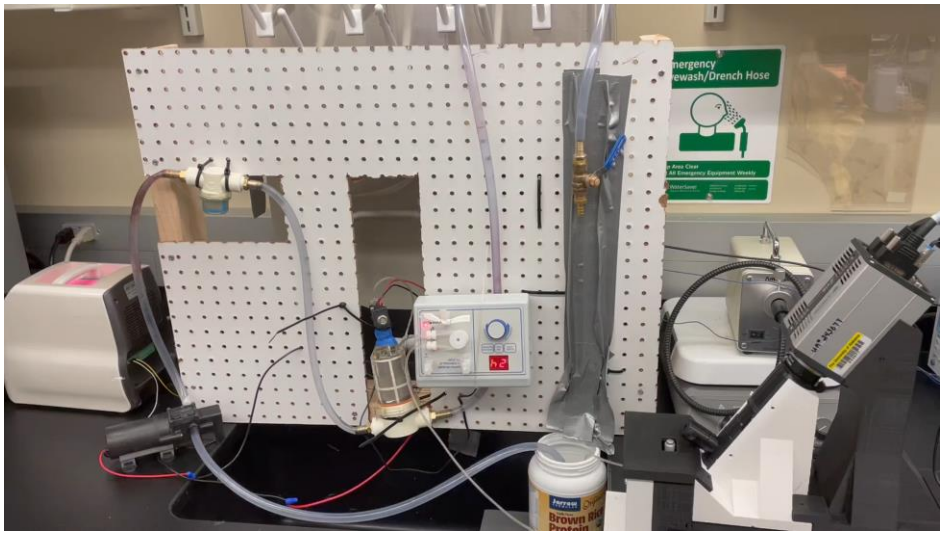


Figure A.3 Initial Test Setup

Figure A.4 Lake Lurleen Field Test

Figure A.5 Initial Warrior River Test Setup

Figure A.6 Michigan Field Test Setup

Figure A.7 Current Deployed System in a Field Test at the Warrior River

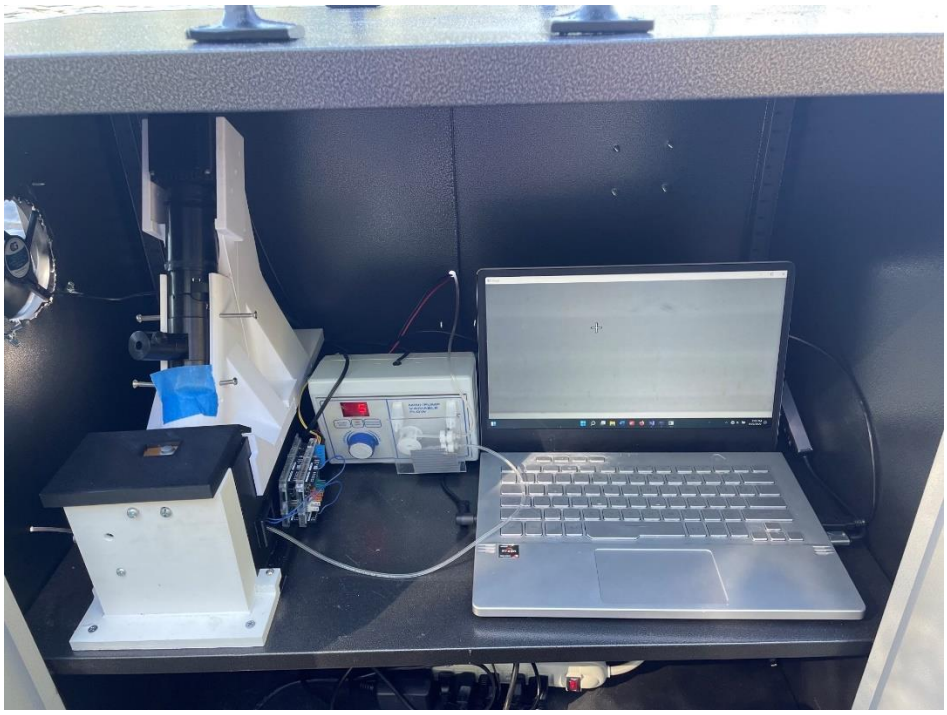


Figure A.8 Top Shelf of Interior Deployed System

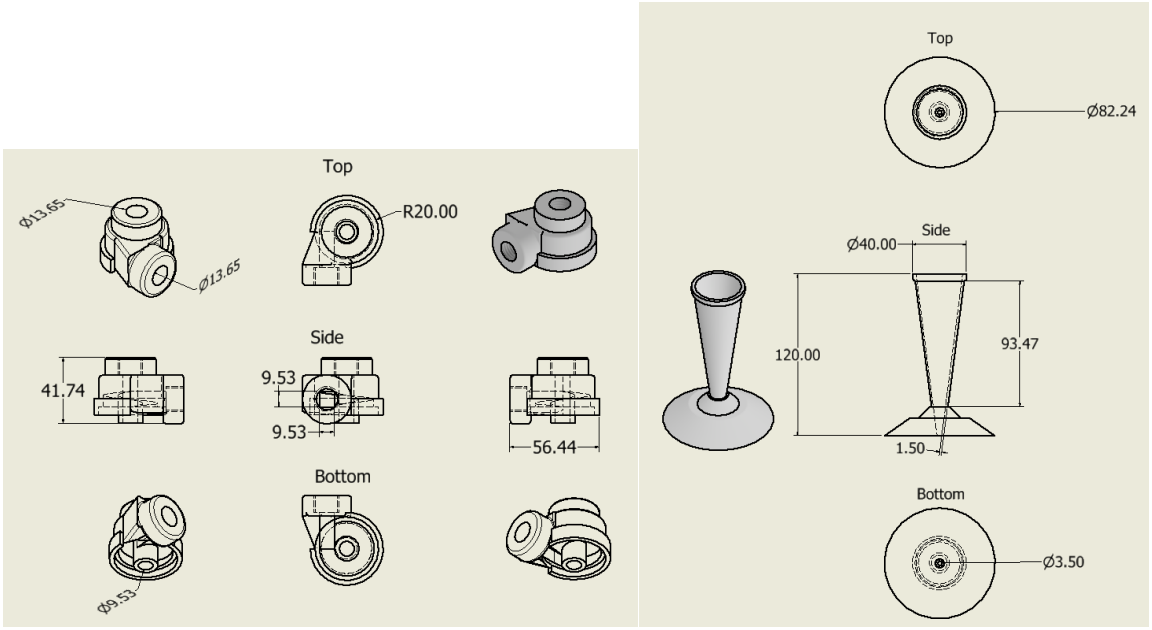


Figure A.9 Cyclone Designs (mm)

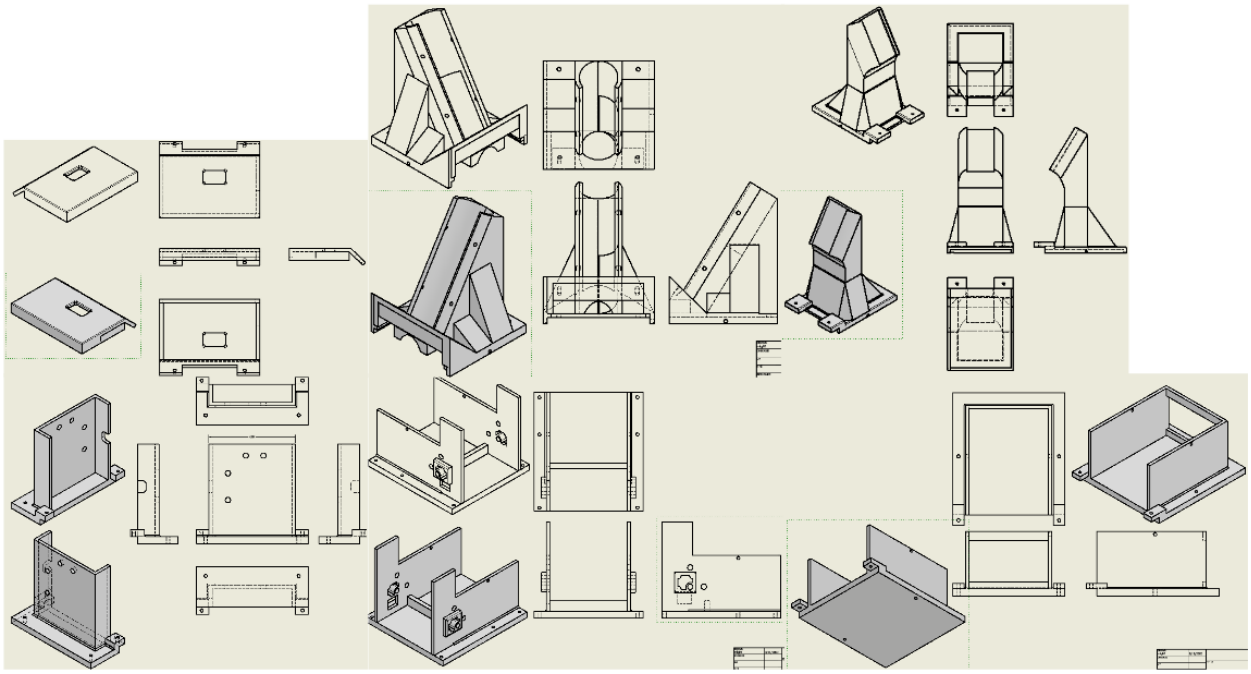


Figure A.10 3D Printed Enclosure Designs

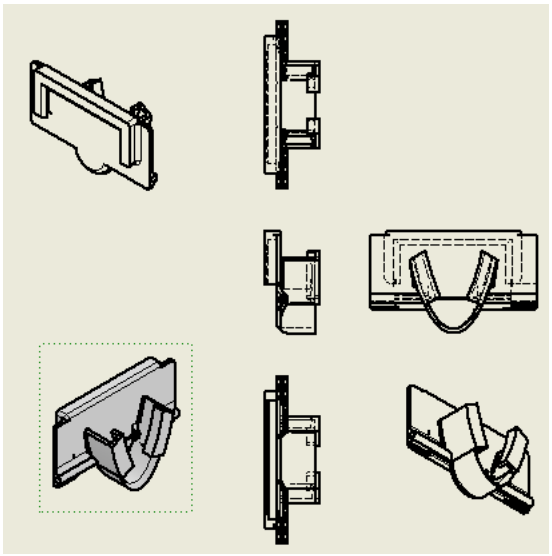


Figure A.11 Tube & LED Holder Design

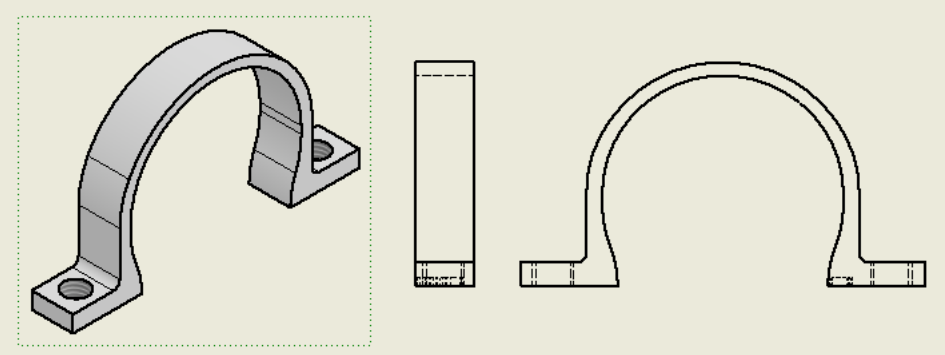
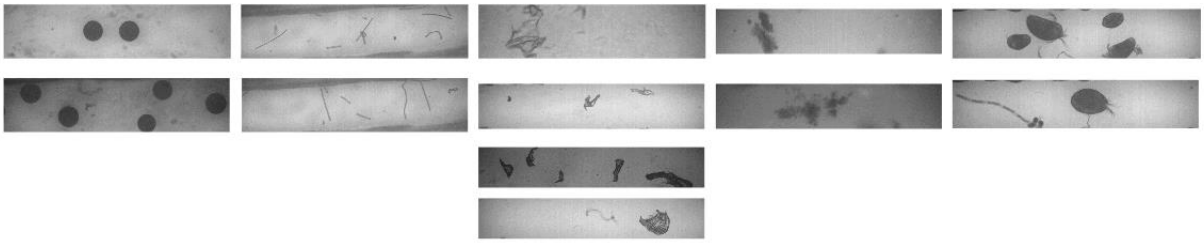


Figure A.12 Strainer Holder Design



FigureA.13 Grayscale Image Examples

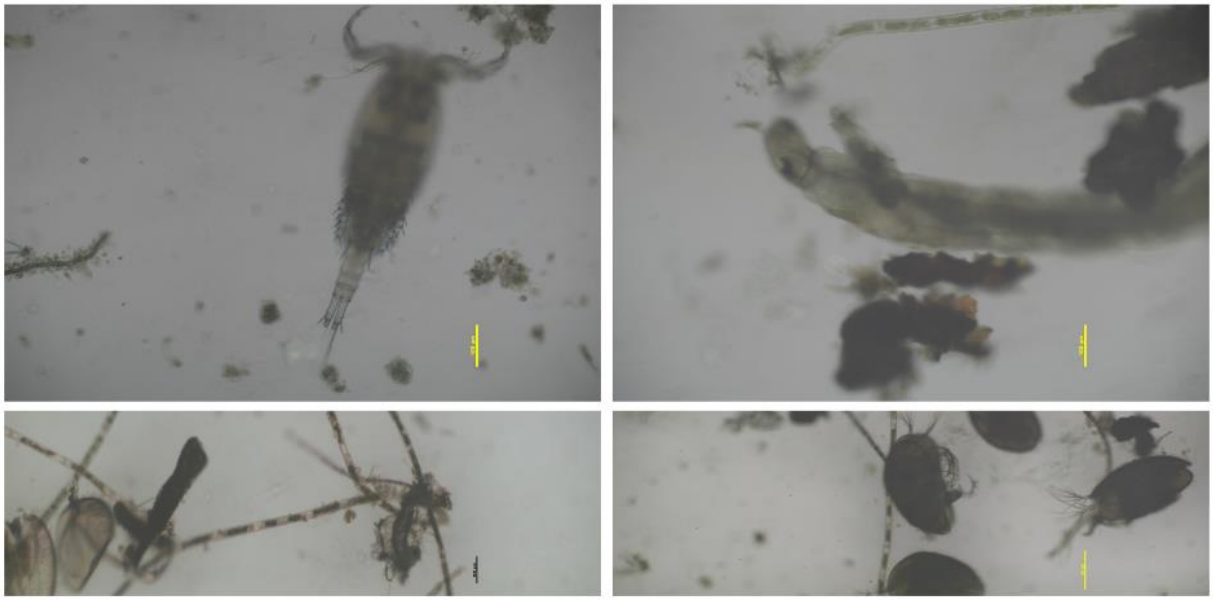


Figure A.14 Color Algae Examples

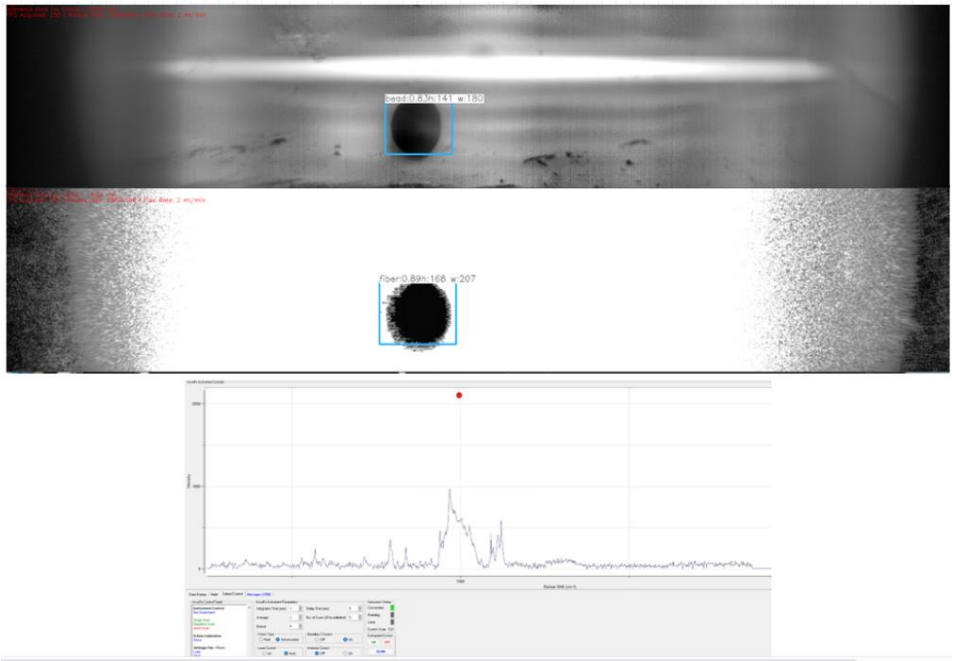


Figure A.15 Detection and Scan Result of PE Pellet

Figure A.16 PP & HDPE Samples Used in Dataset

	HDPE	PP	PET
Color	1025	840	560
Grayscale	1260	250	1168

Table A.1 Number of Total Images Collected for Each Fragment Type

	Fragment	Pellet	Fiber	Dirt	Algae
Color	2425	37	2571	707	720
Grayscale	2678	1178	236	100	156

Table A.2 Number of Total Images Collected for Each Class

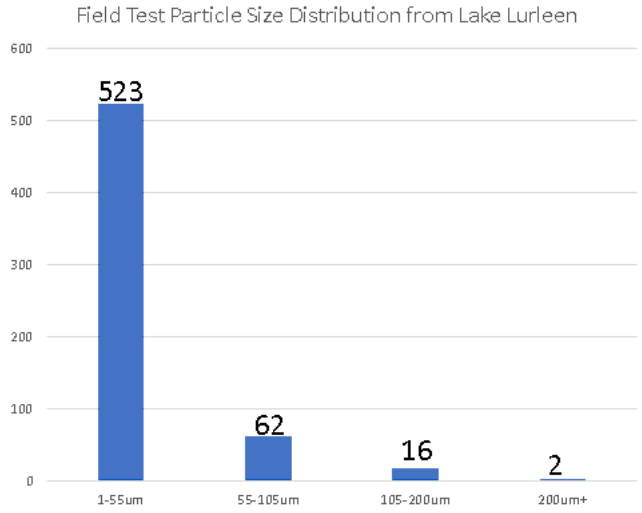


Figure A.17 Lake Lurleen Particle Distribution

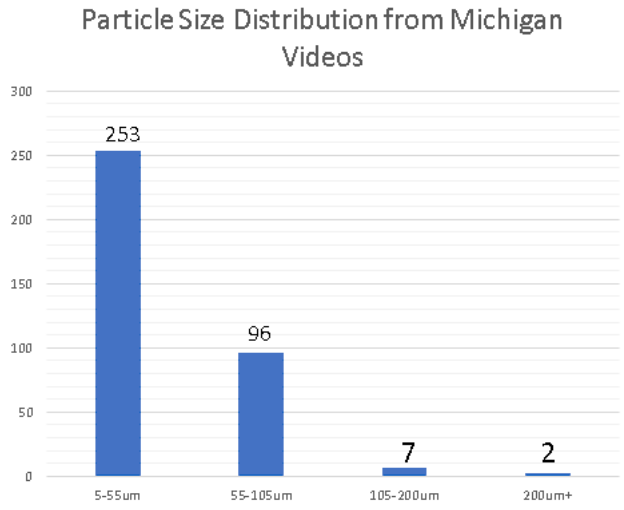


Figure A.18 Michigan Field Test Particle Distribution