Detection of Plastic Pellets on Beach Sand Surface.

Project Report

Reshad Zadran

Aalborg University
Electronics and IT

AALBORG UNIVERSITY
STUDENT REPORT

Electronics and IT
Aalborg University
http://www.aau.dk

Title:
Detection of Plastic Pellets on Beach Sand Surface.

Theme:
Master

Project Period:
Spring Semester 2023

Project Group:
Reshad Zadran

Participant(s):
Reshad Zadran

Supervisor(s):
Rikke Gade

Copies: One

Page Numbers: 66

Date of Completion:
June 1, 2023

Abstract:

This project aims to contribute to the development of a robot that is capable of removing plastic pellets from beaches and coastal environments. The contribution is in the form of an investigation of possible solutions for an autonomous plastic pellet detection system and the development of the system. The system uses a detection algorithm that has been trained on a dataset consisting of RGB images of plastic pellets on beach sand surfaces. Two datasets were used, one with a simple beach environment and the other with a complex beach environment. The complex beach environment was considered complex due to the trash debris, beach material, and other artifacts that could be in that environment. The simple environment contained only sand and plastic pellets. The results from training the detection algorithm on the simple dataset showed that the detector can accurately recognize plastic pellets in a simple beach environment. The training results for the simple dataset were 0.977 for precision, 1 for recall, and 0.995 for mAP. The detection model performed less accurately on the complex dataset with a score of 0.758 for precision, 0.835 for recall, and 0.850 for mAP.

Contents

Preface

Aalborg University, June 1, 2023

Reshad Zadran

Reshad Zadran
<rzadr21@student.aau.dk>

Chapter 1

Introduction

Plastic materials have polluted land masses as well as marine environments. The mass production of plastic products, combined with the irresponsible disposal process of plastic material, has caused plastic products to be leaked into the environment in masses, where the plastic material damage the environment and its inhabitants. The environment can not naturally decompose plastic materials due to the chemical properties of plastic, which makes plastic inherit resistant[1] to degradation. The degradation resistance results in plastic material littering the environment for hundreds of years. Additionally, the inhabitants of the environment have a risk of confusing the plastic material as food, thus eating it and dying or experiencing negative effects because of it. In 2015 Gall and Thompson[2] estimated that 690 marine species have consumed plastic materials in some form, including species that are on the verge of extinction. Apart from being confused as food, plastic in the oceans also washes up on the beaches and coastal environments, where it damages the environment. Cleaning plastic material from the beaches and coastal areas can be difficult due to plastic debris and similar materials are hard to spot. Especially plastic pellets, which are the base for creating plastic products. Plastic pellets are difficult to clean from the beaches and coastal areas due to their small size, and some of them are camouflaged by the sand due to their matching color or by being buried in the sand.

Currently, no solutions exist for removing plastic pellets specifically. Most of the cleaning solutions are focused on removing waste from rivers, oceans, and similar environments, such as Clearbot[3], Wasteshark[4], Mr. Trash Wheel[5], The Interceptor[6]. One thing all of them have in common is that they use a conveyor belt to transport waste into trash bins, that they carry with them. What separates them from each other is how they collect waste. The Wasteshark is remote-controlled by an operator and the operator can decide where the robot should go and what trash it should collect. Clearbot is an autonomous solution that uses machine learning technology to detect and collect. Mr. Trash Wheel and The Interceptor both collect waste by guiding it towards their conveyor belt which uses barriers that waste materials in the river can not pass through. The Interceptor's barriers can be up to 600m long and can go up to three meters deep.

For cleaning the beach there is Bebot[7] which works similarly to a vacuum cleaner. Bebot is a remote-controlled robot that vacuums areas it drives over. It is not a very efficient solution as it is not autonomous and furthermore, it removes the natural beach material from its environment.

This project aims to be a part of the development of an autonomous waste-removing robot. The robot would take advantage of machine learning algorithms to detect plastic and use the location information from the detection to remove only the plastic pellet from the beach.

Chapter 2

Problem Analysis

In this chapter first, the pros and cons of plastic are explained. Afterward, the challenges with plastic pellets are detailed. Next, the initial problem definition is derived based on the challenges with plastic pellets. Thereafter, the related work within plastic detection is explored, and from the research on the related work, a solution to the problem is proposed. At the end, the final problem definition and a conclusion to the chapter are given.

2.1 Problem Description

Plastic has had and still have a huge impact on the world, both positive and negative. The positive effect stems from the flexibility of plastic being able to be turned into many different products. With the invention of plastic, industries such as manufacturing, packaging, beverage, and more were able to use the flexibility of plastic to their advantage and innovate their products completely by mass-producing plastic parts for vehicles, aircraft, technological devices, containers for consumables, furniture, and much more. Figure 2.1 illustrates some of the areas where plastic is used.

Figure 2.1: Plastic is used in, from top left: Aerospace, Construction, Electronics, Packaging, Automotive, Furniture, Healthcare, Military, Energy Generation, and Marine[8].

2.1.1 Growth of plastic production

The negative effect of plastic is in the form of plastic pollution. Studies on plastic pollution [9], [10], [11], [12], [13], [14] shows the main causes to be plastic production, plastic disposability, and plastic recyclability. Figure 2.2 shows the growth of plastic production from 1950 to 2019.

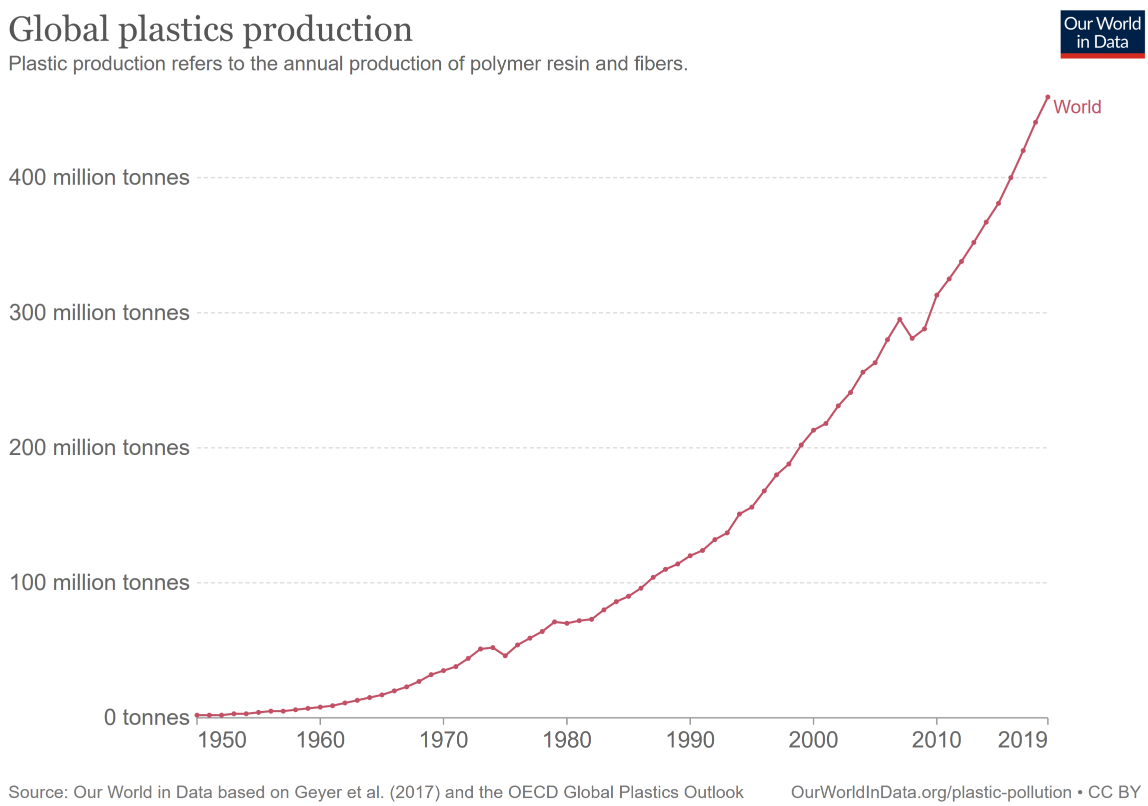


Figure 2.2: Growth of plastic production from 1950 to 2019[9].

From 2010 to 2019 plastic production increased from 270 million tonnes to 460 million tonnes, which is an increase of roughly 70%, in less than a decade[9]. The huge growth overshadowed plastic's ability to be recycled and disposed of. From 1950 to 2015 6.3 billion tonnes of plastic waste was produced, 9% was recycled, 12% incinerated, and 79% stored in landfills or released into the oceans and other natural environments[14].

2.1.2 Disposability and recyclability properties of plastic

Plastic is difficult to dispose of because it can take up to 1000 years[15] to decompose, hence why it is stored in landfills. Storing plastic in landfills is a temporary solution while searching for a complete solution. Landfills are not a complete solution due to the stored plastic being leaked from the landfills into the natural environment, damaging the environment and its inhabitants.

Incinerating all plastic is also not a complete solution, due to the toxins released by the chemical reaction between heat and plastic. Furthermore, airborne toxins from the reaction can be carried by the wind to mountain peaks and other isolated areas, where it affects the natural environment.

Not all plastic can be recycled due to the relationship between heat and plastic. Different types of plastic require different heat levels, hence it would require a more complex recycling system. Additionally, plastic products similar to plastic bags can get in the way of the recycling process by getting wrapped around the machinery, thus forcing the personnel to stop the recycling process to clean the machinery.

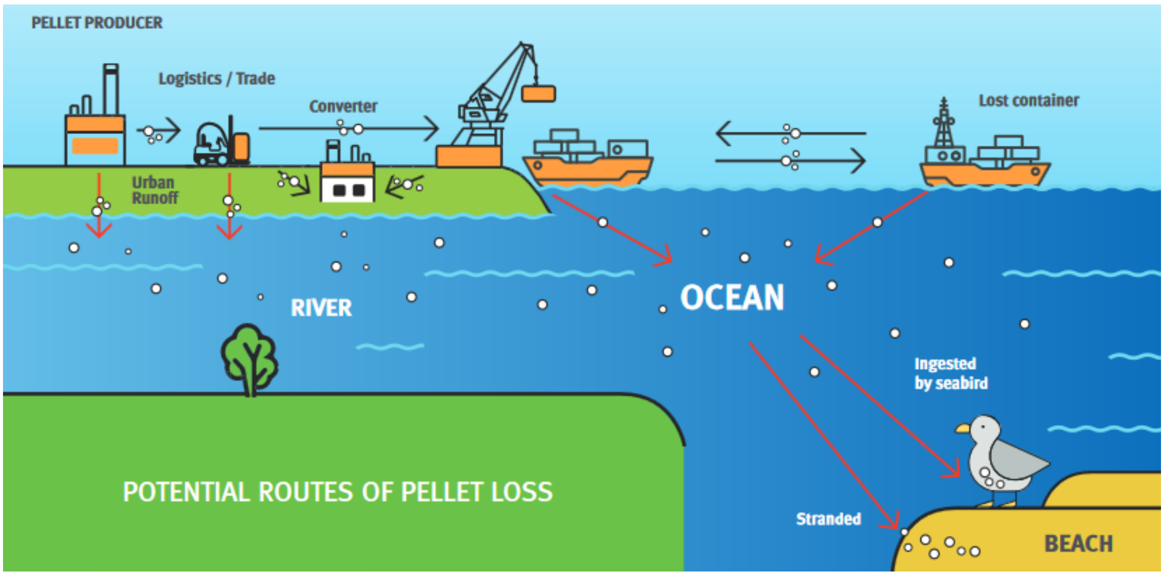


Figure 2.3: Different routes plastic pellet leaks into the enviroment[16].

Plastics' poor disposability and recyclability properties stem from plastic not being biodegradable, due to the primary sources used in manufacturing plastic being crude oil and natural gas. The problem with plastic not only lies in what primary sources are used to manufacture it but also from the beginning of the manufacturing process till the end of it. Figure 2.3 illustrates how plastic pellets leak into the environment from the beginning of the production process till the end.

2.1.3 Manufacturing process of plastic

The manufacturing process[17] of plastic can be simplified into five steps: extraction, refinement, cracking, polymerization, and product shaping. In the extraction step, crude oil and natural gas are extracted from the ground and transported to a refinery. In the refinement step, chemical processes are used to obtain ethane from crude oil and propane from natural gas. In the cracking step, ethane and propane are broken down into ethylene and propylene. In the polymerization step, resins are created by adding a catalyst to turn ethylene into polyethylene and propylene into polypropylene. The seven main different types of plastic created by the cracking and polymerization steps are Polyethylene Terephthalate (PET), High-Density Polyethylene (HDPE), Polyvinyl Chloride (PVC), Low-Density Polyethylene (LDPE), Polypropylene (PP), Polystyrene (PS), others (O). Figure 2.4 shows the different types of polymers used to make various plastic products. In the product shaping step, the resins are turned into plastic pellets by melting, cooling, and cutting them. The plastic pellets are then transported to manufacturers that can shape the pellets into what they desire by using heat to mold the plastic pellets.

POLYMER TYPES	EXAMPLES OF APPLICATIONS	SYMBOLS
Polyethylene Terephthalate (PET)	Fizzy drink and water bottles. Salad trays.	1 PET
High Density Polyethylene (HDPE)	Milk bottles, bleach, cleaners and most shampoo bottles.	2 HDPE
Polyvinyl Chloride (PVC)	Pipes, fittings, window and door frames (rigid PVC). Thermal insulation (PVC foam) and automotive parts.	3 PVC
Low Density Polyethylene (LDPE)	Carrier bags, bin liners and packaging films.	4 LDPE
Polypropylene (PP)	Margarine tubs, microwaveable meal trays, also produced as fibres and filaments for carpets, wall coverings and vehicle upholstery.	5 PP
Polystyrene (PS)	Yoghurt pots, foam hamburger boxes, plastic cutlery, protective packaging for electronic goods and toys. Insulating material in the building and construction industry.	6 PS
Unallocated references	Any other plastics that do not fall into any of the above categories - for example polycarbonate which is often used in glazing for the aircraft industry.	7 O

Figure 2.4: Overview of the different types of plastic that can be produced[18].

The transportation step in the manufacturing process is where the majority of plastic gets lost to the sea. Tonnes of plastic, including plastic pellets, are commonly transported in containers via ocean carriers. The World Shipping Council (WSC) estimated that on average 1629[19] containers were lost at sea each year in the period between 2008 and 2021. Containers lost to the sea are one of many ways plastic ends up in the ocean. Jambeck et al.[20] estimated that in 2010 12.7 million metric tonnes of plastic entered the ocean, where some of it washes up on the coasts and beaches. Plastic pellets have been sighted on the coasts and beaches of Texas[21], Greece[22], Brazil[23], India[24], New Zealand[25], Lebanon[26], China[27] and many more countries. Because of the small size of plastic pellets, removing them is difficult. The difficulty lies in spotting where the plastic pellets are, as it can be quite challenging to find the plastic pellets when standing and looking down. Additionally, picking up or trying to spot plastic can be a tedious task due to the person cleaning must squat down to look for and collect the pellets. Figure 2.5 shows a close-up photo of plastic pellets and plastic debris lined up in a sand environment. Some of the pellets in figure 2.5 can be hard to spot when not looking up close.

Figure 2.5: Plastic pellets are lined up on the left side and plastic debris on the left side.

2.2 Initial problem definition

Plastic waste can be found in many places, they have different shapes and can be of different plastic types. This project specifically aims to analyze a detection solution for plastic pellets found on the sand surface of the beaches. The scope of the project is limited due to time constraints and to provide an in-depth report of plastic pellets found on the beaches. The initial problem definition is based on the problems mentioned in the problem description and is as follows:

Investigate a solution for an autonomous plastic pellet detection system. The system must be capable of finding plastic pellets in a surface sand environment.

2.3 Related Work

This section entails the existing work within the field of plastic detection. First, Current solutions within plastic detection are detailed. Afterward, based on the research from the works within the field, a method is chosen for further development and research.

2.3.1 Detection methods

Within the field of plastic detection, the most common approaches for detecting plastic are machine learning with a combination of RGB (red, green, blue color spectrum) images or near-infrared spectroscopy (NIR). Ultraviolet (UV), X-ray, and sonar can also be used but is not as common. The RGB, NIR, and UV approaches are the main focus of this project due to data obtainability and accessibility issues. The related work within each approach will be explored in depth to build a greater understanding of the methods.

2.3.2 Near-infrared and ultraviolet light method

Near-infrared spectroscopy is commonly used for plastic classification to sort plastic types into recycled and non-recyclable types [28], [29], [30], [31], [32], [33]. [34]. Polymers for each plastic type have a different molecular structure, which results in different NIR absorption and reflectance patterns. The patterns are used as features to train machine learning algorithms, which enables them to detect plastic and classify plastic types. Similar to NIR, ultraviolet light methods take advantage of the molecular structure of plastic to get spectral features that can be used to classify plastic material. There are few publications on plastic classification with UV light[35], [36], [37] compared to NIR. The works within both methods lack publication of the NIR and UV method used for detection purposes only, hence classification with NIR and UV is investigated further to see if the methods used within that field can be applied for detection-only approaches. The methods of Zinchik et al.[29], Zhu et al.[32], Neo et al.[34], and Gies et al.[37] are researched further as they are using machine learning approaches for plastic resin type detection. The publications are explored in greater depth to investigate if the machine learning approach could be used for detection-only methods.

Accurate Characterization of Mixed Plastic Waste Using Machine Learning and Fast Infrared Spectroscopy

Zinchik et al.[29] proposes to use the framework shown in figure 2.6. The input to the framework consists of a NIR spectrum that is converted into a one dimension vector (1D) data object. Gramian Angular Field (GAF) method is used to transform the 1D vector into a two-dimensional (2D) data object. The GAF methods used are Gramian Angular Summation fields (GASF) and Gramian Angular Difference fields (GADF). The GAF methods are used to improve classification accuracy. The GAF matrices are fed into their Convolution Neural Network (CNN) which outputs what class the plastic material belongs to.

A five-fold cross-validation method is for training and testing their CNN with a ratio of four to one, respectively. Their results varied with the size of the input matrix to the CNN, with an input size of 25×25×2, an accuracy of 99.5% is achieved. With an input size of 100×100×2, an accuracy of 100% is achieved.

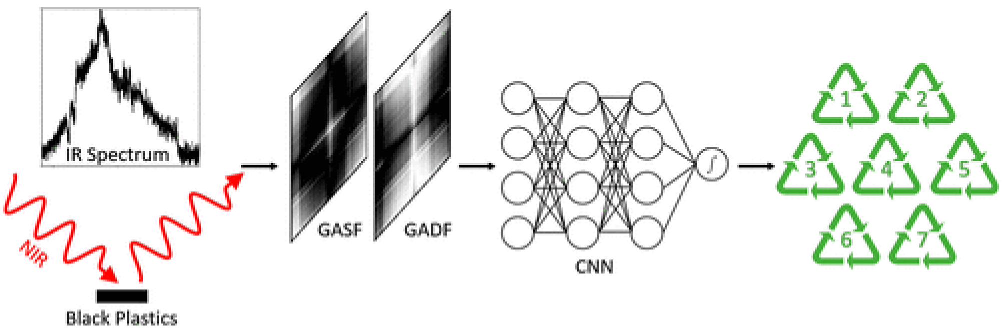


Figure 2.6: The framework proposed by Zinchik[29].

Plastic solid waste identification system based on near-infrared spectroscopy in combination with support vector machine

Zhu et al.[32] presents a support vector machine (SVM) based method to classify plastic type. For data collecting, a NIR optical fiber spectrometer is used. The spectral data is enhanced by normalizing the data, taking the first derivative of the data, and smoothing the data. After the spectral data is enhanced, principal component analysis (PCA) is applied to the improved spectral data. PCA is applied to speed up data analysis. Radial Basis Function (RBF) is used to construct an SVM model that classified the improved spectral data.

186 samples were used for training and validation. For validation, a 10-fold cross-validation approach and grid search was used. The PCA-SVM model achieved an accuracy of 97.5%.

Deep learning for chemometric analysis of plastic spectral data from infrared and Raman databases

Neo et al.[34] propose a deep learning architecture named PolymerSpectraDecisionNet (PSDN) for plastic classification. The architecture is structured as a decision tree and consists of CNNs, residual networks, and inception networks. PSDN splits the classification into two networks. The first network takes spectral data of plastic material as input and classifies the data as recyclable or non-recyclable. The second network uses the data classified as recyclable and further classifies the data into individual plastic types. The PSDN architecture is illustrated in figure 2.7.

Their dataset is split into 75% for training, 15% for validation, and 10% for test. Additionally, for testing a 10-fold cross-validation is used. PSDN achieved an accuracy of 96.7%.

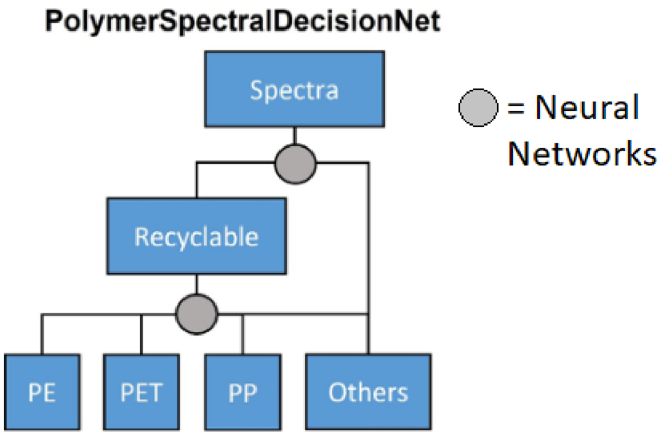


Figure 2.7: The PolymerSpectraDecisionNet architecture proposed by Neo et al.[34].

Investigating the Potential of UV-excited Photoluminescence Spectroscopy for the Identification of Plastics

Gies et al.[37] investigated the possibility of using UV light to identify the plastic type. They use a TOPTICA TopWave266 frequency-quadrupled cw-Nd:YAG laser which operates at 266 nm. The laser is used to excite plastic material. The spectra from the UV-excited material are recorded by an OceanOptics HR4000 spectrometer.

They conclude that it is possible to distinguish between each plastic material that has been illuminated with UV light.

2.3.3 RGB method

The RGB method does not rely on data from a spectrometer, but instead on images of plastic in the red, green, and blue color spectrum. The images are used by different classification and detection algorithms to detect plastic and classify plastic types[38], [39], [40], [41], [42], [43], [44]. The work of Bobulski et al.[38], Kim and Kubanek[39] are explored deeper to gain a greater insight into the different applications of the RGB method. Additionally, the work of Wolf et al.[43] is looked into further to understand how plastic size affects the detection of plastic litter.

An Innovative Automated Robotic System based on Deep Learning Approach for Recycling Objects

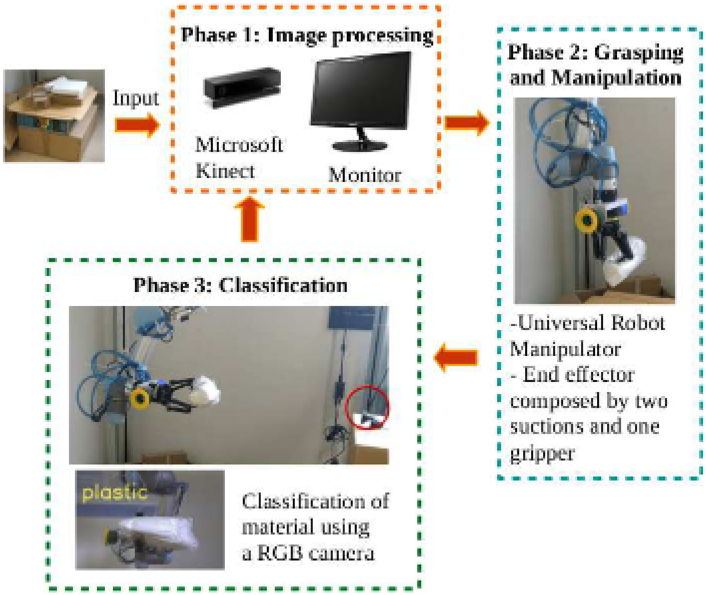


Figure 2.8: Complete overview of the system proposed by Kim and Kybanaek[39].

The system presented by Kim and Kubanek[39] is illustrated in figure 2.8. The main four parts of the system are image processing, grasp detection, material classification, and motion planning. From the camera, a point cloud of the object is generated and used by the manipulator to get the right path and grasp for the object. Afterward, a modified LeNet-5 is used to recognize if the object is plastic or carton, depending on what class the object is then the object is placed in a box belonging to that class.

The modified LeNet-5 was trained for 250 epochs. It achieved an accuracy of 99% during training and 96% during testing.

Deep Learning for Plastic Waste Classification System

Bobulski et al.[38] propose a system that consists of an RGB camera, a microcomputer, and an air jet. The stream from the RGB camera is used as input for CNN running on the microcomputer. The CNN classifies the plastic type and actives the air jet to push the plastic material into a bin, belonging to that type.

Their CNN achieved an accuracy of 97.43%.

Machine learning for aquatic plastic litter detection, classification and quantification (APLASTIC-Q)

Wolf et al.[43] propose a system named the aquatic plastic litter detection, classification and quantification system (APLASTIC-Q). The system consists of a plastic litter detection network (PLD-CNN) and a plastic litter quantifier network (PLQ-CNN). Both networks are fed RGB images of plastic litter. The plastic objects can vary in size but most of them look small due to the images being collected by aerial surveys, but the size does not affect the detection capabilities of the CNN networks. Figure 2.9 shows a plastic litter region. The PLD-CNN estimates plastic litter in an area, while the PLQ-CNN detects how many of each plastic object is in the image region. They have 18 different plastic object classes such as plastic cups, plastic bottles, small plastic bags, and more.

The APLASTIC-Q system achieved a precision score of 0.77, recall score of 0.77, and F1-score of 0.77.

Figure 2.9: Image of a plastic litter region that is used as input to the APLASTIC-Q system[43].

2.3.4 Section conclusion

This section researched the existing methods for plastic classification and detection. NIR, UV, and RGB methods were explored further for a deeper understanding of the field. The NIR and UV method was commonly used for classification of plastic types, while the RGB method was used for classification and detection, as seen in the works of [43], [39]. Additionally, Wolf et al.[43] showed that the plastic content of RGB images taken six meters from the ground can be detected and classified. Based on the insight from this section, an approach similar to that of Wolf et al.[43] is deemed to be sufficient for detecting plastic pellets on the sand surface of the beach. The approach would entail using close-up (less than one meter) RGB images of plastic pellets as input for a detection network to localize and recognize the plastic pellets.

2.4 Proposed solution

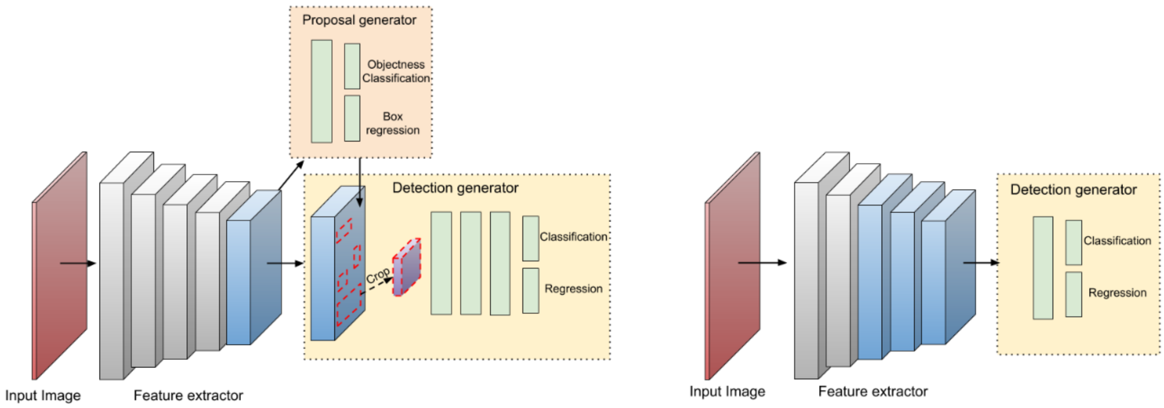
The RGB method is a viable solution for this project due to the characteristics of plastic pellets and the environment they are in. Plastic pellets are different from plastic debris. Plastic debris can consist of any plastic material that has been ripped apart or torn into smaller pieces, as seen in figure 2.5. Plastic debris is also prone to degradation, whereas plastic pellets are much less prone to degradation due to the pellets being in their solid form which has yet to be molded into a product. Furthermore, looking at the plastic pellets up close makes spotting them less difficult due to the solid form and the clear color of the plastic pellets. Additionally, the natural beach environment does not contain objects similar to plastic pellets, hence plastic pellets stand out and are more visible in the natural beach environment. It is important to notice the difficulty of being able to spot the characteristics of plastic pellets depends heavily on distance, as looking down from shoulder height makes spotting plastic pellets a very difficult task compared to looking from knee height. Close-up RGB images of plastic pellets are a viable input for machine learning algorithms to detect them due to the characteristics mentioned.

The proposed solution consists of creating a dataset of plastic pellets, annotating the dataset, training a detection algorithm on the plastic pellet dataset, and evaluating the performance of the detection algorithm. This section explores the related work within object detection in order to choose an optimal detection algorithm for finding plastic pellets.

2.4.1 Object detection

Object detection is a very popular field with a lot of existing work[45], [46], [47], [48], [49], [50]. The evolution of the field accelerated with the addition of convolutional neural networks. Modern object detectors commonly have either a one-stage architecture or two-stage architecture. One-stage detectors do object classification and bounding-box regression without a region proposal network (RPN), which enables faster inference. Applications, where the environment is rapidly changing, require fast inference to not lose important information. An example of such an application is a system for self-driving cars, if inference does not happen fast then the application might not detect nearby cars, which could result in a fatal accident. Classic one-stage detectors trade speed for performance, they have faster inference, but lower accuracy compared to two-stage detectors. Traditionally, two-stage detectors use an RPN to obtain high accuracy and because of that, they have slow inference. Figure 2.10 shows the two architecture types.

The plastic pellet detector must have fast inference, as it is intended to be used for beach surveys, in real-time. Additionally, the fast inference time allows for the detector to run on mobile devices, hence the cleaning task of plastic pellets can be done without the need for strong computational devices. Therefore, the related work focuses on one-stage detectors, where the works of Redmon et al.[50] and Liu et al.[49] are looked deeper into.



(a) Two-stage Faster R-CNN **(b)** One-stage RetinaNet

Figure 2.10: Neural network architecture with RPN and with no RPN.[51].

You Only Look Once: Unified, Real-Time Object Detection

Redmon et al.[50] propose the object detector You Only Look Once (YOLO). YOLO is continuously being improved, As of 2023 YOLOv8 has been released. YOLO does not use any region proposal methods instead, it uses a single CNN to simultaneously predict multiple bounding boxes and class probability for the boxes, all in one evaluation. The image inputted to the CNN is split into grids. A bounding-box and confidence score are predicted for each grid of the image. The grids with the highest score are chosen as the location where an object could be.

SSD: Single Shot MultiBox Detector

Liu et al.[49] presents a method named Single Shot MultiBox Detector (SSD) for object detection. Similar to YOLO, SSD uses a single CNN and does not use an RPN. SSD takes in an image as input and feature maps are extracted from the image. Each feature map location produces a fixed collection of bounding-boxes with a confidence score for all object categories. The bounding-box with the highest confidence of an object is chosen to be the location of said object. Figure 2.11 gives an overview of the method.

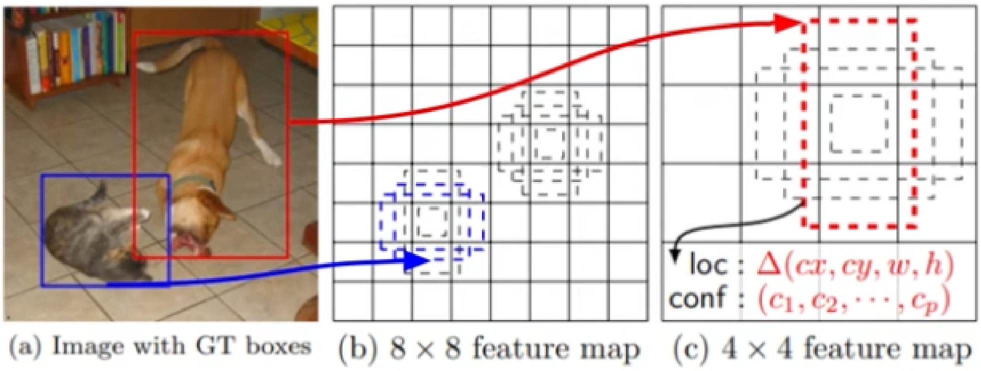


Figure 2.11: Object detection in SSD[52].

2.4.2 Section conclusion

This section proposed a solution for the detection of plastic pellets. The solution entailed using the RGB method combined with an object detection algorithm. One-stage detectors were explored deeper, as they have fast inference and are commonly used for real-time (30 frames per sec (fps)) applications. Based on the research of one-stage detectors, YOLO is chosen to be the most optimal choice, as it offers fast inference, good accuracy, continuous development, and good integration with Python.

2.5 Final problem definition

Can images of beach sand surfaces from an RGB camera be used to train a YOLOv5 algorithm, for an autonomous detection system of solely plastic pellets on the beach sand surface?

2.5.1 Requirements

Requirements for the model are set to ensure the detection algorithm works as intended and has optimal performance, in order to actually use it in real-life systems. The precision, recall, and mean average precision (mAP) values are set as requirements for the training phase and the inference phase of the model. The model must achieve the requirements set in both phases in order to pass.

1. *An RGB camera must be used for data collection.*

   This project uses the RGB method for plastic pellet detection, thus the dataset acquired must be in RGB color space in order for the dataset to be used for a plastic pellet recognition algorithm.

2. *Plastic pellets of different colors must be detectable.*

   Plastic pellets of different colors are found in the natural beach environment. The colors include blue, red, yellow, black, and possibly more. The plastic pellet detector must be capable of finding different colors of plastic pellets in order to clean the beach.

3. *Plastic pellets that are translucent must be detectable.*

   Translucent plastic pellets must also be cleaned from the beach, hence the detector must be capable of finding them.

4. *The detection algorithm must have a precision of at least 0.70.*

   A precision score of at least 0.70 ensures that at least 70% of the plastic pellets are classified correctly.

5. *The detection algorithm must have a recall of at least 0.90.*

   A recall score of at least 0.90 ensures that at least 90% of the plastic pellets are found. Recall has been given a higher priority than precision, as it is important to remove as many plastic pellets as possible from the beach.

6. *The detection algorithm must have a mAP of at least 0.85.*

   A mAP of 0.85 ensure that the detector's performance is optimal and can be relied on to be implemented in a real system.

2.5.2 Delimitations

In order to provide an in-depth report of the plastic pellet problem and a solution to it, certain delimitations are set, so the project can deliver on its goals within the limited time frame.

1. *The solution focuses only on the detection of plastic pellets.*

2. *Plastic debris is not seen as plastic pellets.*

3. *Plastic pellets on only the surface of the sand must be detectable.*

4. *Reflection and occlusion is not accounted for.*

2.6 Conclusion

This chapter started by analyzing the positives and negatives of plastic. The cons were elaborated further by explaining the rapid plastic production, plastic pollution, and plastic manufacturing process. In the plastic production explanation, plastic pellet was highlighted as a source of plastic pollution in all stages of their production, especially transport. Afterward, the problems with removing plastic pellets were detailed, and an initial problem definition was created based on the challenges with the removal. The next section explored the existing solutions for the detection and classification of plastic. From the research of existing work, a solution was proposed. The solution entails using a camera to create a dataset of plastic pellets, annotate the dataset, train a You Only Look Once (YOLO) algorithm to detect the plastic pellets, and evaluate the performance of the algorithm. Afterward, the final problem definition was defined based on the solution section, and the requirements for the solution were outlined.

Chapter 3

Methods

This chapter explains the methodology of this project. First, a general overview of all the components of the project is given. Afterward, data acquisition and data annotation are detailed. Next, machine learning and convolutional neural networks are explained in greater depth. In the end, YOLOv5 and its building blocks are detailed further.

3.1 System overview

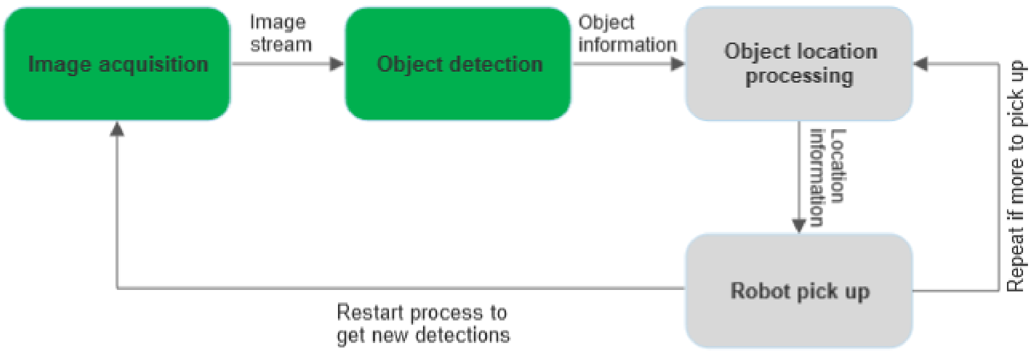


Figure 3.1: A general overview of the entire system. The gray boxes are not within the scope of this project.

Figure 3.1 shows a general overview of the entire envisioned system, for cleaning plastic pellets off the beaches. The greyed-out boxes are not within the scope of this project, whereas the green boxes are. The first step in the process is to obtain an image stream of plastic pellets on the beach sand surface, which is done by an RGB camera mounted on the robot. The camera feed is processed by an object detection algorithm in order to find plastic pellets at the current robot location. Next, the localization information from the object detector is used to get the position of the plastic pellets in the robot frame from the camera frame. Thereafter, the location information of the plastic pellets is fed to the robot gripper in order to pick up the plastic pellet from the beach. Depending on how many plastic pellets have been detected in the current area, the location passing and pick-up process is repeated till there are no more detections in that area. The robot then moves to the next area and repeats the entire process.

3.2 Dataset

3.2.1 Dataset quality

There are various factors involved in making a good dataset, some of the factors are size, Representativeness, and labeling quality.

Dataset size

The dataset size is depended on the application. With some applications, a few hundred images are enough for the detecting algorithm to learn the features of the objects in the images, hence it can recognize the objects in images that it has not seen before. Meanwhile, other applications require thousands of images to perform the same task.

Representativeness of dataset

Representativeness ensures that the dataset contains an object that is of importance to the application and that the dataset has a diverse representation of that object. Meaning different colors, shapes, angles, and more of the object, hence the object in the dataset becomes a reflection of the actual object in the real-world. A dataset that is representative improves the capabilities of a detection algorithm when performing object recognition in real-world systems.

Labeling quality of dataset

Labeling quality is a very important factor for object detection algorithms. It is the base for what object the detection algorithm learns to recognize. Being consistent and precise when annotating a dataset ensures reliable class and bounding box predictions from an object detection algorithm.

3.2.2 Race for Oceans Foundation dataset

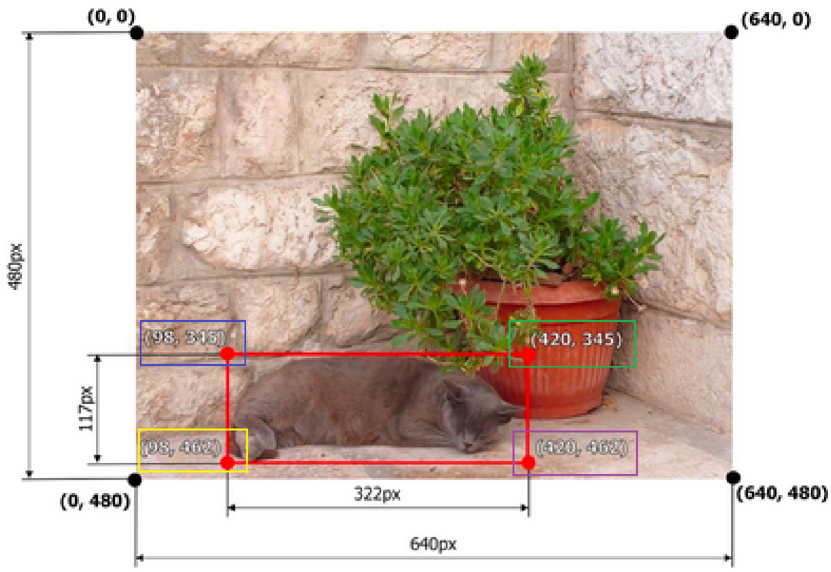
Over 15000[53] images were collected from a trip along the west coast of Denmark. The route stretched 456 kilometers (km), starting in Rømø and ending in Skagen. Along the route, 40 different beaches were visited to take pictures of plastic pellets. The pictures were taken with a JAI GO-5100C-USB camera with a KOWA LM12HC lens. The camera settings were set to auto as the environment was changing dynamically in different locations. The field trip was conducted by the Race for Oceans Foundation[53]. Figure 3.2 shows some of the images taken on the field trip.

Figure 3.2: Images of plastic pellets on the beach surface, obtained from a field trip along the coast of Denmark[53].

The Race for Oceans Foundation[54] was established in 2019 with its base in Løkken, Denmark. The foundation aims to develop AI, robot, and drone solutions for getting rid of plastic in the oceans and coastal environments. The solutions are developed in cooperation with the Alexandra Institute and Aalborg University.

3.2.3 Annotation

Annotation is the process of labeling the objects in an image. The objects are labeled in order for the computer algorithm to learn the patterns of the object, hence becoming capable of recognizing the specific object, in images that it has not seen before. Which objects to label depends on the application, i.e., a car recognition application would label all cars in an image. The label for object detection consists of the class of the object and its location. The class and location information is typically in row format as a text file, where the first value in the row indicates what class the object belongs to and the remaining values are the coordinates of the bounding box encapsulating the object. A bounding box is a rectangle used to mark an object of interest in an image. YOLO uses four values to represent a bounding box, the annotation format of values are: x center, y center, bounding box width, and bounding box height. The four values are normalized relative to the image. Figure 3.3 shows the described annotation format of YOLO and the calculation process. In figure 3.4 an annotation example of the Race for Oceans Foundation dataset is shown.



`[((420 + 98) / 2) / 640, ((462 + 345) / 2) / 480, 322 / 640, 117 / 480]`

↓

`[0.4046875, 0.840625, 0.503125, 0.24375]`

Annotation format: 0 0.4046875 0.840625 0.503125 0.24375

Figure 3.3: Illustration of the YOLO bounding box calculation and annotation format. The first index in the annotation format is the class of the object, in this case, class 0 is cat[55].

Figure 3.4: The image shows an annotation example from the Race for Oceans Foundation dataset. The plastic pellets are annotated with a red bounding box.

3.3 Machine learning

This section gives an introduction to the field of machine learning, in which convolution neural networks like the chosen YOLOv5 is a branch of.
Machine learning (ML) uses computer data and algorithms to imitate human learning intelligence. The imitation is based on how humans would solve the task ahead without being explicitly programmed for the task. The imitation is made possible by creating a model that learns the patterns in the data. Another approach that is used is a reward-punishment method, the model can be rewarded for getting the task right or punished for getting it wrong, thus the model is trained to learn how to solve a certain task. The task can be to recognize an object, perform a physical action, write text, or something similar. The learning methods commonly used to train the models are supervised learning, unsupervised learning, and semi-supervised learning[56].

Supervised learning

Supervised learning is commonly used for object detection. The machine learning algorithm learns the pattern of the object by using a labeled dataset of the object for training the algorithm. The algorithm is then able to recognize the object in images that it has not seen before. If the algorithm trained on a labeled dataset of dogs and cats, then the algorithm would learn their features and learn which feature belongs to what object, thus being able to recognize cats or dogs. The downside of supervised learning is that it requires a labeled dataset, if such does not exist then it has to be made, which can be very time-consuming.

Unsupervised learning

Unsupervised learning is often used for applications where an event is vastly different from the rest, hence why it is also known as outlier detection. In the field of anomaly detection, unsupervised is a common approach for training an algorithm, due to how rare abnormal events occur and the rarity of recorded abnormal events, such as jaywalking, fleeing, etc. In contrast to supervised learning, Unsupervised learning does not use any labeled data, hence it is faster to create a dataset for an unsupervised algorithm, but in return the prediction of the algorithm is less accurate.

Semi-supervised learning

Semi-supervised learning is created to overcome the downside of unsupervised learning. The downside of unsupervised learning is its prediction accuracy. Semi-supervised learning improves accuracy by having a small set of the dataset labeled when training the algorithm.

3.4 Convolutional Neural Networks

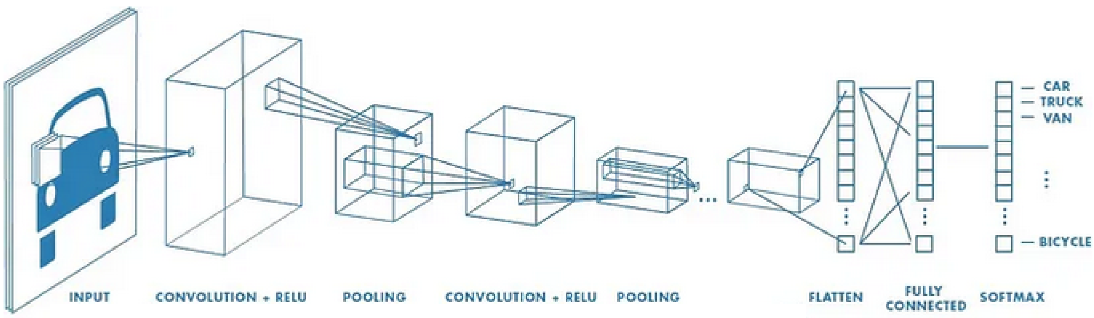


Figure 3.5: Classic convolutional neural network architecture[57].

Convolutional neural network (CNN) is a part of the machine learning field. Figure 3.5 illustrates the traditional blocks of the CNN architecture. CNNs ability to recognize patterns makes it a powerful tool for image recognition and processing. The general architecture of CNNs consists of convolution layers, pooling layers, and fully connected layers. CNNs also use non-linear activation functions at the output of convolution and fully connected layers[58].

Convolution layer

The task performed by the convolution layer[58] entails using the dot product operation between the input matrix and the filter matrix. The input matrix consists of pixel values. The filter matrix is also known as a kernel and the values of the kernel are known as weights. Typically, the weights of the kernel are initially random but are adjusted/learned during the training of the network. The size of the kernel is its receptive field and it can vary, YOLOv5 uses a combination of 1x1, 3x3, and 6x6 kernels.

The convolution operation superimposes the kernel on the pixel values within the receptive field of the kernel and outputs the dot product at that specific location to a new matrix called the feature map. The operation is repeated by sliding the kernel over all the pixel values, that are within the receptive field of the kernel. Figure 3.6 illustrates the convolution operation.

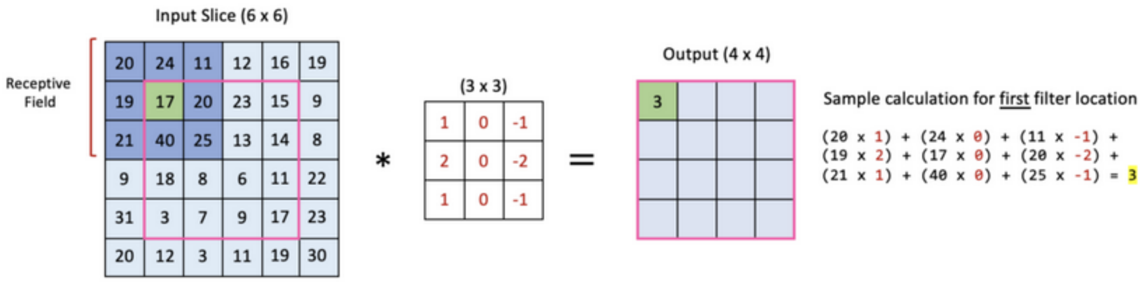


Figure 3.6: Result of the convolution operation. The 6x6 matrix contains pixels of an image. The 3x3 matrix is the kernel and the 4x4 matrix is the result of the convolution operation. The 3x3 dark blue field is the receptive field of the kernel and the center of the kernel is marked with green. The pink rectangular outline indices the pixels within the receptive field of the kernel[59].

The sliding of the kernel can be done in different strides, a stride of one is equal to sliding the kernel by one pixel at a time, while a stride of two skips one pixel. The convolution operation reduces the resolution of the input image, especially with a stride size greater than one. In contrast, with padding the resolution of the image is preserved, by adding pixel values of zero around the border of the image. Whether a network uses padding or increased stride size or both depends on the application[58]. For YOLO speed is important, hence different stride sizes are used to reduce computational power. YOLOv5 uses a combination of stride sizes of one, two, and three.

Batch normalization

Batch normalization[60] is typically used after the convolution layer. Batch normalization is a standardization of feature maps, that is scaled and offset. The standardization process changes a feature map's mean and standard deviation to a new value. The values of the feature map are then scaled (multiplied) by alpha and offset (summed) with beta. The parameters for scaling (alpha) and offsetting (beta) are learned during the training of the network. Equation 3.1 shows the formula used for the batch normalization process in CNNs.

$$x_{BN} = (\frac{(x - \overline{x})}{\sigma} * \alpha) + (\beta) \tag{3.1}$$

$$x_{BN} = \text{Batch normalized value, } x = \text{Pixel value}$$

$$\overline{x} = \text{Mean of the feature map, } \sigma = \text{Standard deviation of the feature map}$$

$$\alpha, \beta = \text{Variables learned during training}$$

Non-linear activation function

Generally, non-linear functions[58] after batch normalization. The complex shape patterns of images can not be described with linear combinations, but can be approximated with non-linear combinations. The nonlinearity is introduced to CNNs via a non-linear activation function. The most commonly used non-linear activation functions are rectified linear unit (ReLU), sigmoid, and hyperbolic tangent (tanh). Each of the functions operates differently, ReLu zeros negative values and returns positive values as they are. Sigmoid collapses a value to be within the range of zero and one. High values are close to one and small values are close to zero. Tanh works similar to the sigmoid function, but instead of squashing the input within the range of zero and one, it does between the range of minus one and one. Figure 3.7 shows the three non-linear activation functions mentioned.

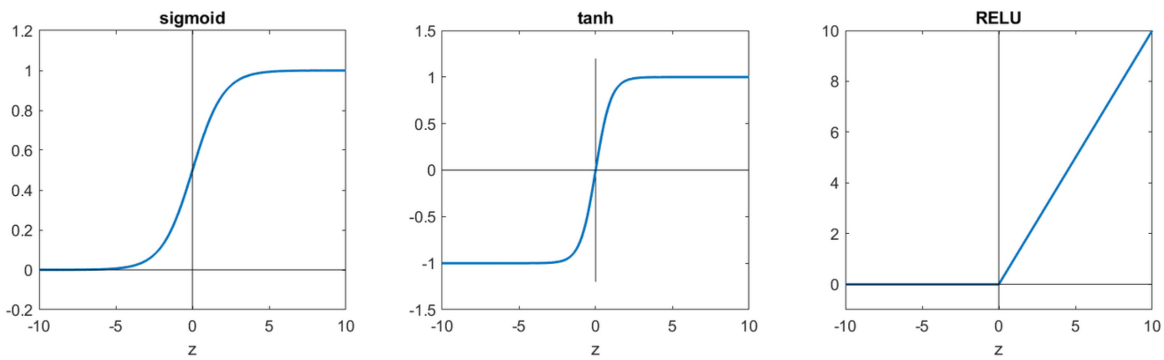


Figure 3.7: Overview of the most commonly used non-linear activation functions[61].

Pooling

Pooling[58] is often used after the non-linear activation function. Pooling looks at a region of the feature map and picks a value of that region for a new feature map. The value picked depends on what pooling operation the application needs, but the most common pooling approaches are max pooling and average pooling. Max pooling takes the highest value in the region, while average pooling takes the average value of the region. Pooling reduces spatial resolution while preserving the most important features. Reduced spatial resolution, increases the computational speed of the network.

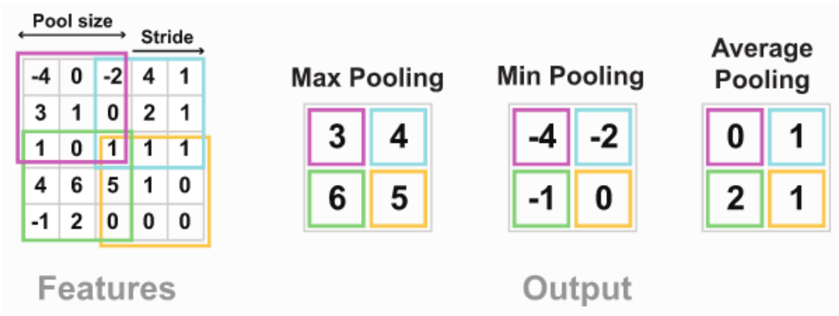


Figure 3.8: Illustration of the three most common pooling operations[62].

Upsampling layer

Upsampling[63] can be used to increase the spatial resolution of an image. In the case of YOLOv5, upsampling is used to look for objects in three different image scales. A scale for small objects, medium objects, and large objects. Upsampling of the image is done by a nearest-neighbor approach. The approach upsamples a feature map by a factor of m. The feature map is upsampled by inserting m minus one row/column after each row/column of the feature map. The new columns and rows are signed with a value of zero. Afterward, the convolution operation is used on the upsampled feature map (zero-inserted image) and the nearest-neighbor kernel. Zero-padding is used on the zero-inserted image in order to not lose resolution and to be able to find the neighbors of the border pixel values. The upsampling process is shown in figure 3.9.

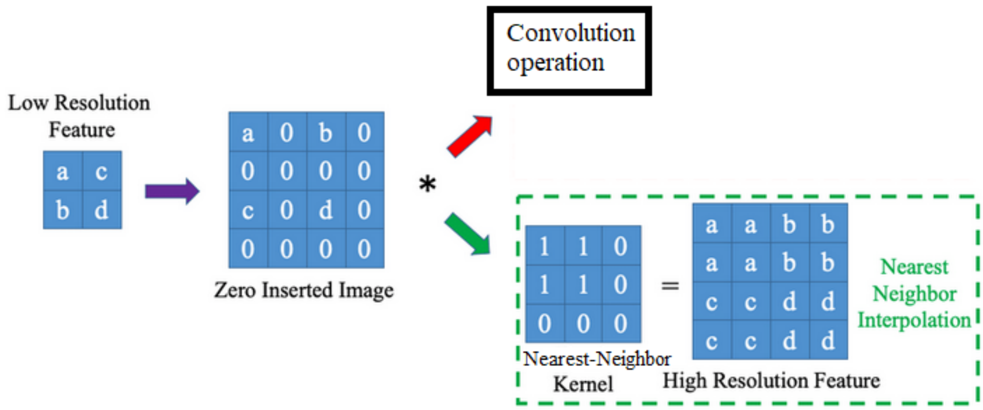


Figure 3.9: Illustration of the nearest-neighbor upsampling technique[63].

Fully connected layer

The fully connected layer[58] is the last layer in traditional CNN architecture. It consists of an input layer, one or more hidden layers, and an output layer. Each layer has a certain amount of neurons. A neuron takes in multiple inputs and multiplies the individual inputs with their weights, then sums the results, next it adds bias to the sum, afterward the result is fed to the non-linear activation function which then passes its output to the next layer. In a fully connected layer, all neurons in each layer are connected to all the neurons in the previous layer and the next layer. The input to the first layer (input layer) is a one-dimensional feature vector. The output layer uses a softmax function to output the class probabilities for the image.

Hyperparameters

Hyperparameters[64] are variables that are set before training the network. They affect the speed and the accuracy of the network. YOLOv5 has 28 hyperparameters. The hyperparameters commonly changed are learning rate, batch size, and epochs.

The learning rate is used by the optimizer of the network to determine the step size needed for the network to minimize the loss function of the network. The loss function indicates how wrong the network is. YOLOv5 uses stochastic gradient descent (SGD) as its default optimizer for minimizing the loss function.

The batch size determines how many samples must be processed by the network before the weights of the network are updated. Larger batch sizes require more memory but speed up the training of the network due to a larger amount of samples that can be processed at a time.

An epoch is one forward and backward pass of the whole dataset through the network. Iterations and epochs are closely related to one another, as an epoch consists of a certain amount of iterations, depending on the batch size. One iteration is one forward and backward pass of one batch of samples.

General functionality of CNN

The operations performed by a CNN can be split into a feature learning part and an object recognition part[57]. In the first part, features of the input are learned through convolutions, non-linear functions, and pooling. The first convolution layers learn low-level features such as lines, corners, edges, etc. while the deeper convolution layers learn mid- to high-level features such as object parts and whole objects. Figure 3.10 shows the different features learned in the feature learning phase.

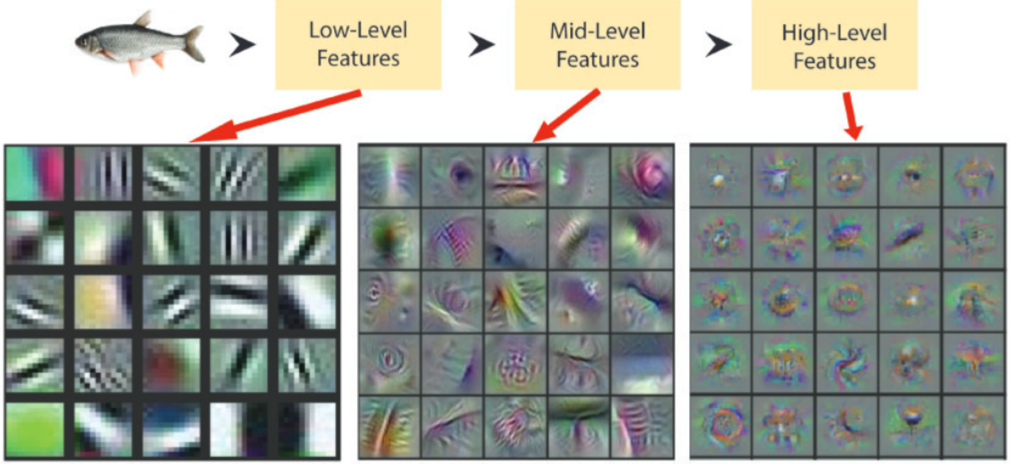


Figure 3.10: Illustration of features learned by CNN as an image is processed deeper into the network[65].

In the object recognition part, the output from the feature learning phase is flattened into a column vector and fed into the fully connected layer. The fully connected layer uses a softmax function to get the probabilities of the features, belonging to a certain class. The class with the highest probability is chosen as the object. Non-linear functions and backpropagation enable the fully connected layer to learn non-linear combinations of the features, hence becoming able to differentiate between the features and assign a high or low probability for a class, depending on the similarity it has with the processed image.

Commonly, machine learning algorithms use a known pre-trained CNN architecture (backbone) for feature learning. Pre-trained backbones are convolution neural networks that are trained on very large publicly available datasets. Training one from scratch would consume a lot of time and require a device with very high computational power. Using pre-trained backbones speeds up development, as they have already been trained and their effectiveness is known. YOLOv5 is used in this project, and it uses CSPDarknet53 as its backbone. Some well-known backbones/CNN architectures are AlexNet[66], LeNet[67], VGGNet[68], GoogLeNet[69], ZFNet[70], and ResNet[71]. Figure 3.11 illustrates the performance and accuracy of various backbones.

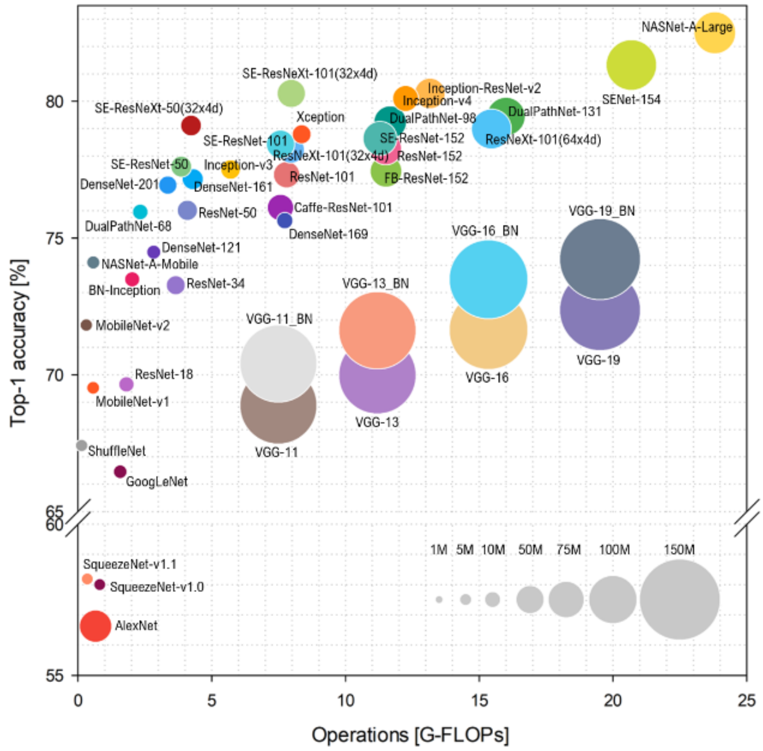


Figure 3.11: Accuracy and floating point operations per second (FLOPS) of various CNN backbones. The figure shows that computationally expensive backbones have very high accuracy e.g. SENet-154, while less computationally expensive backbones do not have as high accuracy but have very good performance[72].

3.5 YOLOv5

YOLOv5 is a one-stage object detector, figure 3.12 shows a simple overview of the YOLOv5 architecture, consisting of three blocks: a backbone (CSPDarknet53), a neck (SPPF and PANet), and a head (YOLOv3). Each block serves a specific purpose.

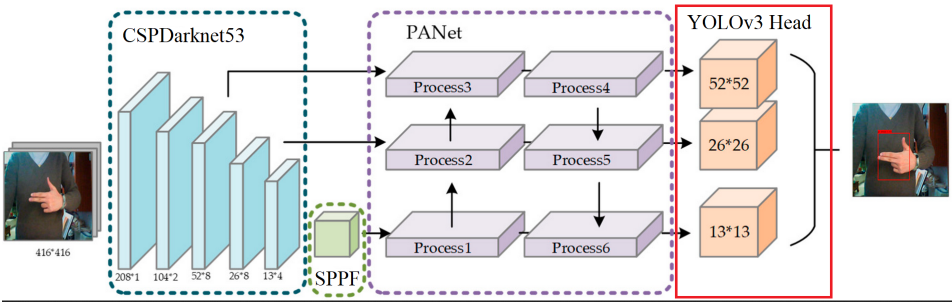


Figure 3.12: Simplified overview of the main building blocks of YOLOv5[73].

Backbone

The backbone of YOLOv5 is a pre-trained CSPDarknet53 network. The backbone is used to extract features of images, reduce the spatial resolution of the image, and increase feature resolution. CSPDarknet53 is the Cross Stage Partial (CSP) strategy applied to Darknet53. Darknet53[74] is a CNN with 53 convolutional layers, 23 of the convolutional layers have residual blocks to overcome the vanishing gradient problem. The CSP[75] strategy used by YOLOv5 entails splitting the feature map of the base layer into two parts. Part two goes through convolutions (Conv), batch normalization (BN), and Sigmoid-Weigthed Linear Units (SiLU) activation function (ConvBNSiLU layer), then goes through bottleneck layers, and at the end, it gets concatenated with part one, which did not go through the ConvBNSiLU layers, figure 3.13 illustrates the process. Bottleneck layers are used to increase the depth of the network while also reducing the number of parameters and matrix multiplications, hence the network becomes better at generalization while also having fast inference time. Overall, the CSP strategy enhances the learning capabilities of the network by enabling better gradient flow. Additionally, the strategy also reduces computation effort, which increases inference speed. For real-time algorithms, fast inference time is very important.

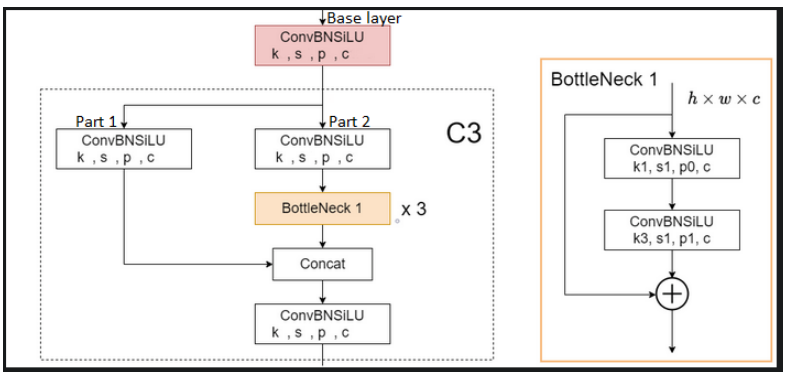


Figure 3.13: The CSP strategy used in combination with bottleneck blocks[76].

Neck

The model neck uses a modified Path Aggregation Network (PANet) and a modified Spatial Pyramid Pooling (SPP), to improve the generalization of the model to objects of various sizes and scales.

PANet[77] is an improvement of the feature pyramid network (FPN) architecture. Deep CNNs with FPN architecture risk losing low-level features due to the depth of the network. PANet addresses the problem by connecting low-level features to high-level features with a bottom-up path augmentation technique, to ensure that the information from the low-level features can be used to localize objects. YOLOv5 uses the PANet architecture with the addition of the SPP strategy, bottom-up path augmentation strategy, and CSPNet strategy to improve the information flow of the network and its feature recognition capabilities. Figure 3.14 shows the PANet architecture, linking low-features to high-level features.

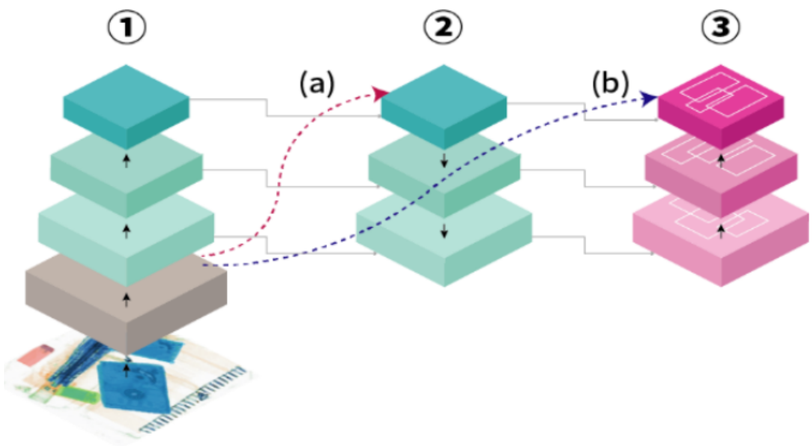


Figure 3.14: Low-level features (a) and (b) getting linked to high-level features with the PANet bottom-up path augmentation technique. (1) contains the low-level features, (2) medium-level features, and (3) high-level features[78].

With the SPP[79] technique, CNNs can use images of different sizes as input without having to warp or crop the image, thus not losing image information or resolution. The method is applied to a CNN by adding an SPP layer on top of the last convolution layer. The SPP layer is added before the fully connected layer because the fully connected layer requires a fixed-length input. SPP can generate fixed-length outputs by dividing a feature map into different bins, then pooling the bins, and afterward concatenating the output from the pooling operation to a fixed-length representation that can be fed to the fully connected layers. Figure 3.15 illustrates the process, where the same feature map is split into three bins, the first bin is the entire feature map, then the feature map is split into four bins, and lastly, the feature map is split into 16 bins. The max-pooling operation is performed on each of the bins, and then they are concatenated. YOLOv5 uses Spatial Pyramid Pooling Faster (SPPF), SPPF is the SPP method but with fewer floating point operations per second (FLOPS)[80].

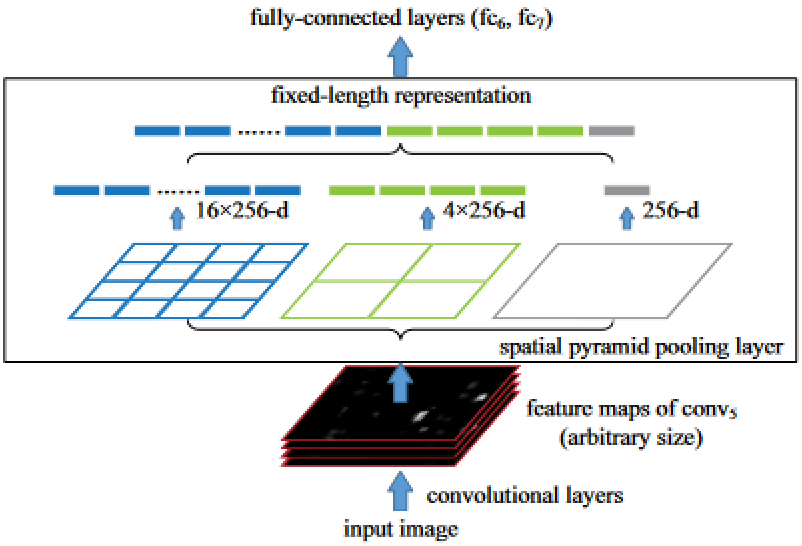


Figure 3.15: Illustration of the Spatial Pyramid Pooling approach[79].

Head

The object classification and bounding box regression are done by the model head. Classification and bounding-box regression take place on three image scales: low resolution, medium resolution, and high resolution. Each scale is responsible for detecting objects of that scale, i.e. low scale detects small objects. The detection is done by using a grid-based approach. The method entails dividing the image into grids. A grid consists of one or more pixels in the image. Each grid predicts three anchor boxes for what it sees. The prediction tensor contains the box coordinates, objectness score, and class scores of the grid. Four box coordinates are used to predict the bounding box: the center coordinates of the anchor box(x, y), the height of the image, and the width of the image. The objectness score is the likelihood of the grid seeing an object. The class score is the probability of an object belonging to a certain class. Non-maxima suppression (NMS) is used to remove overlapping bounding boxes to get a final bounding box for an object.

Complete network

Figure 3.16 shows a complete in-depth overview of the YOLOv5 network architecture. An image fed into the network first goes through the backbones ConvBNSiLU layers (Conv), CSPBottleneck layers (C3), and an SPPF layer. The fractions next to different layers describe the change in image size, after the first Conv layer, the image is half of its original size, and so on. The neck takes the input from the SPPF layer and upsamples it. Additionally, the PANet architecture is used when the upsampled data is concatenated with its respective feature level in the backbone, to ensure feature loss is minimized. For example, when the image gets upsampled to one-sixteenth of its original size, then it gets concatenated with the last C3 block which contains the richest features at that level. This process is repeated multiple times in the neck for better information flow. The head takes the output from the neck after the concatenation at $\frac{1}{8}$ image size, $\frac{1}{16}$ image size, and $\frac{1}{32}$ image size. At each scale, a tensor containing the batch size, height, width, anchor boxes, and prediction tensor is outputted. The length of the prediction tensor is 85, four attributes for box regression, one for objectness score, and because the network is trained on the COCO dataset, it has 80 class probabilities. Additionally, the head uses NMS to remove overlapping bounding boxes for the final output.

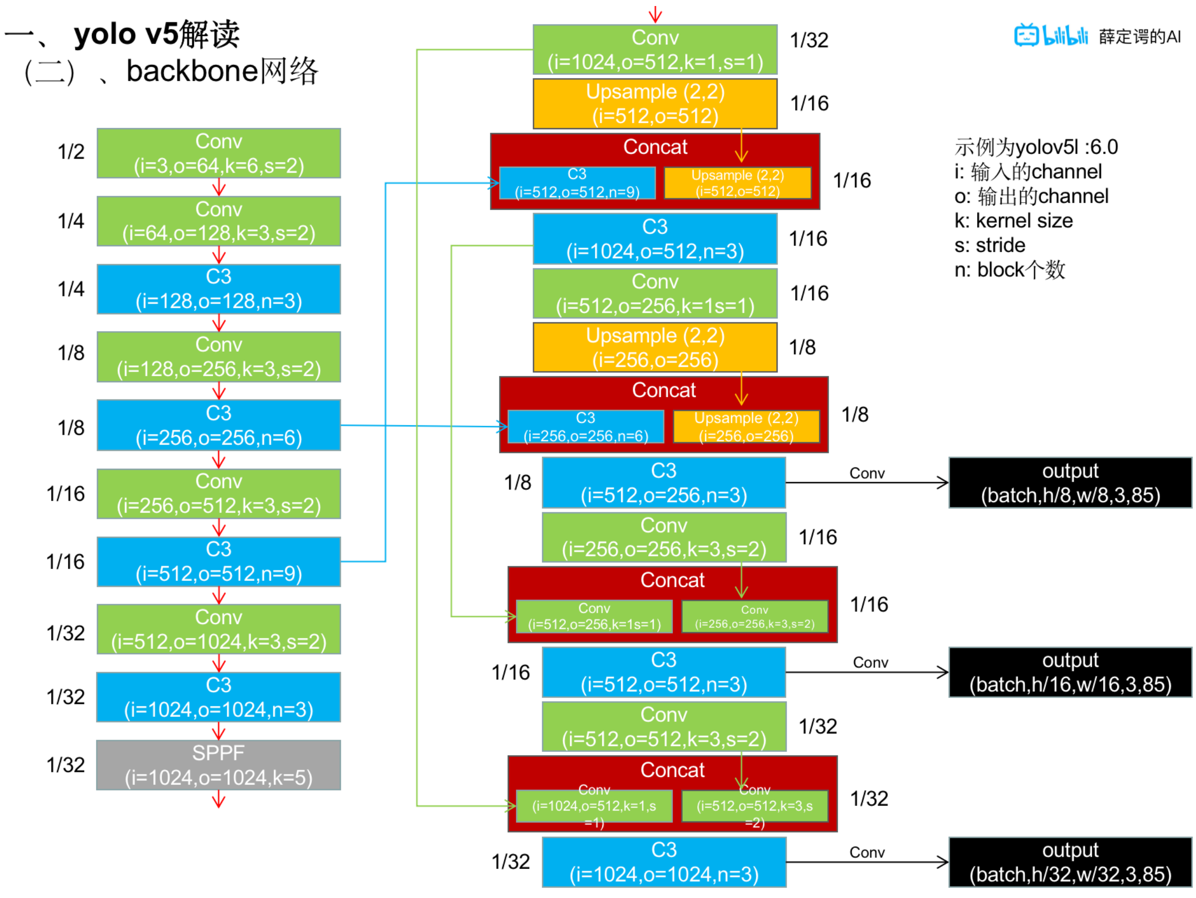


Figure 3.16: A complete overview of the building block used for YOLOv5[81].

Chapter 4

Implementation

This chapter details the implementation of various components used in the development of the final solution. First, the system the solution is developed on is explained. Next, the building blocks of the detection system are elaborated on. At the end, the results from a sanity check of the detection algorithm are detailed.

4.1 System

The system used for developing, training, and evaluating the detection algorithm is an Ubuntu cloud platform. The cloud platform offers great flexibility by being able to connect to it from remote locations, and light computational devices can connect to it, to perform heavy computational work. The cloud server has a powerful graphics processing unit (GPU) and additional random-access memory (RAM), which makes it a good choice for training and testing convolutional neural networks.

4.1.1 Cloud platform

Cloud platforms offer access to various services such as data storage, computing resources, remote access, and more. The cloud service provider (CSP) hosts the services at a remote data center and access to their services is granted by them to subscribers of the service. Aalborg University (AAU) offers such a service, called Strato. Strato is a cloud computing platform, where a user can tailor a cloud instance to meet their needs. Each cloud instance can be initiated with a desired operating system (OS) and a flavor. OSes such as Ubuntu 20, Ubuntu 18, Arch Linux, and more can be chosen. Additionally, a flavor has to be picked for the instance. Depending on what flavor is chosen, different resources such as disk space, RAM, amount of Virtual Central Processing Units (VCPUs), and GPU become available to the cloud instance. This project uses a gpu.A40-large flavor that offers 20 VCPUs, 64-gigabyte ram, 100-gigabyte disk size, and an NVIDIA A40 GPU.

Strato cloud instances can only be created by AAU employers and students. Upon login to the Strato website with an AAU mail, a cloud instance is created in the images tab as shown in figure 4.1.

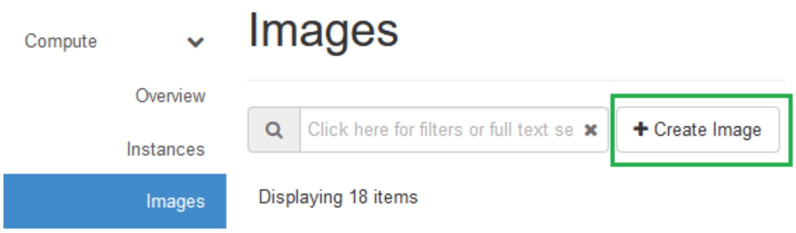


Figure 4.1: Side panel of Strato.

Upon creating an image, the user gets prompted with a window, where the created instance is further customized. The customization includes naming the instance, picking an OS for the instance, choosing a flavor for the instance, choosing a communication protocol for the instance, and lastly generating an authentication key for the instance. Afterward, the instance is connected to via the chosen communication protocol. In order to establish a connection to the instance, the authentication key must be included when connecting to the internet protocol (IP) address of the instance. This project uses an Ubuntu 20.04 instance with a gpu.A40-large flavor and the secure shell protocol (ssh) for communication.

The remote instance is made more interactive by turning it into a remote desktop, by using the X2Go application. X2Go enables a server-client relationship between a host and a user. In the case of the Strato instance, the remote instance becomes the X2Go server, where Light DM graphical user interface (GUI) is installed together with the X2Go server. On the user side, the X2Go application is installed and executed. The X2Go application prompts a window where the IP address, login, authentication key, and session type for the remote instance is specified, thus a connection can be established to the instance. Figure 4.2 shows the X2Go application window and what information it uses.

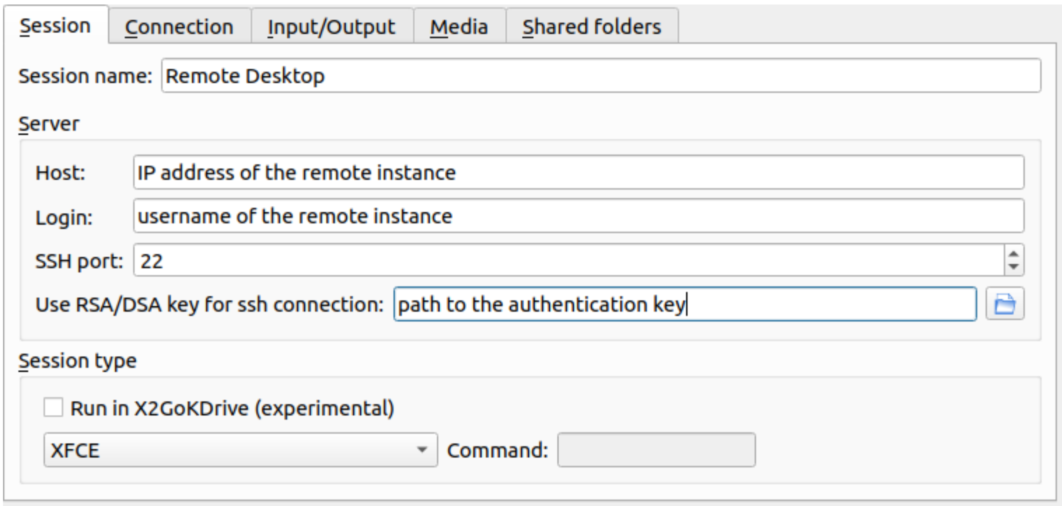


Figure 4.2: X2Go client panel for setting up a remote desktop.

4.2 Detection

The first step in the detection process of plastic pellets is to annotate the images of the plastic pellets. Two datasets of plastic pellets have been acquired, one dataset was made by the Race to Ocean Foundation as described in section 3.2.2, while the other dataset was made from a laboratory setup.

The laboratory setup consists of a plastic tray with sand in it and plastic pellets dropped on top of the sand surface, the plastic pellets are collected from different beaches. A Samsung Galaxy S21 Ultra 5G was used to capture images of the laboratory setup, the camera specification can be found at[82]. Figure 4.3a shows the plastic tray with sand in it. Figure 4.3b and figure 4.3c show images from the laboratory dataset. Similar to the Race for Oceans dataset described in section 3.2.2, the laboratory (controlled) dataset must be annotated In order to use the dataset for training the detection algorithm.

(a) **(b)** **(c)**

Figure 4.3: Image 4.3a shows the setup, 4.3b and 4.3c are images from the controlled dataset.

4.2.1 Annotation tool

Computer Vision Annotation Tool (CVAT) is used for labeling the images. CVAT is an open-source[83] annotation tool widely used for annotating datasets in the field of object detection, image classification, and image segmentation. CVAT offers many features that speed up the annotation process. Some of the features include semi-automatic annotation, Interpolation of shapes, various annotation export formats, and more.

In this project, CVAT is installed on the Windows OS. The installation of CVAT requires Windows Subsystem for Linux (WSL2), Docker Desktop for Windows, Git for Windows, Google Chrome, and Ubuntu 20.04. Most of the required applications can be downloaded from the Microsoft Store. After having installed the required modules, Ubuntu 20.04 is opened from the Windows menu, and in the prompted terminal the commands in listing 1 are executed. Afterward, CVAT is accessed via localhost:8080 in Google Chrome and used for annotating.

```
#Clone the CVAT repository
git clone https://github.com/opencv/cvat
cd cvat
#Run docker
docker compose up -d
#install CVAT
CVAT_VERSION=dev docker compose up -d
#Register superuser
sudo docker exec -it cvat_server bash -ic 'python3 ~/manage.py createsuperuser'
```

Listing 1: Installation process of CVAT

4.2.2 Pytorch

Various Python libraries are used for deep learning, and PyTorch is one of the most popular among them. PyTorch is based on the Torch deep learning libraries combined with Python's high-level programming language, allowing for rapid development and testing of machine learning algorithms. Key features of PyTorch are high integration with Python, tensor computation, dynamic graph computation, Python support, and more.

Installing PyTorch is done through their site[84]. The installation can be customized to fit desired specifications. The choices for installation are the PyTorch version, operating system, installation package, programming language, and whether to use CPU or GPU. After having chosen the desired options an installation command is generated, the command can be executed in the terminal to start the installation. For this project, PyTorch build version 1.13.0 is used. PyTorch is installed on the Linux OS with the pip package.

The chosen programming language is Python and CUDA version 11.7 is used as the GPU platform.

4.2.3 YOLOv5

Ultralytics is the developer and maintainer of YOLOv5. They are an AI software company based in the United States of America. Their goal is to provide advanced AI tools for solving complex real-world problems. Their flagship product is YOLOv5, which has been made open source and is available on their GitHub[85]. The Ultralytics YOLOv5 is widely used and known for its speed, accuracy, and ease of use.

Training and evaluating a custom dataset is done by first creating a workspace for the implementation, then cloning the YOLOv5 repository into the workspace. The cloned repository has a requirements.txt file that contains a list of the necessary packages for the implementation of YOLOv5. The required packages are installed with pip. The next step is to prepare the dataset, which is done by randomly splitting 80% of the dataset into a training folder, 10% into a validation folder, and 10% into a test folder. Additionally, a label directory is created that contains a folder for the annotations of the training set and another folder for the annotations of the validation set. The last step, before training, is to specify the path to the dataset and the classes within the dataset, which is done by creating a dataset.yaml and specifying it there. The final structure of the entire directory and the contents of the dataset.yaml file is shown in figure 4.4

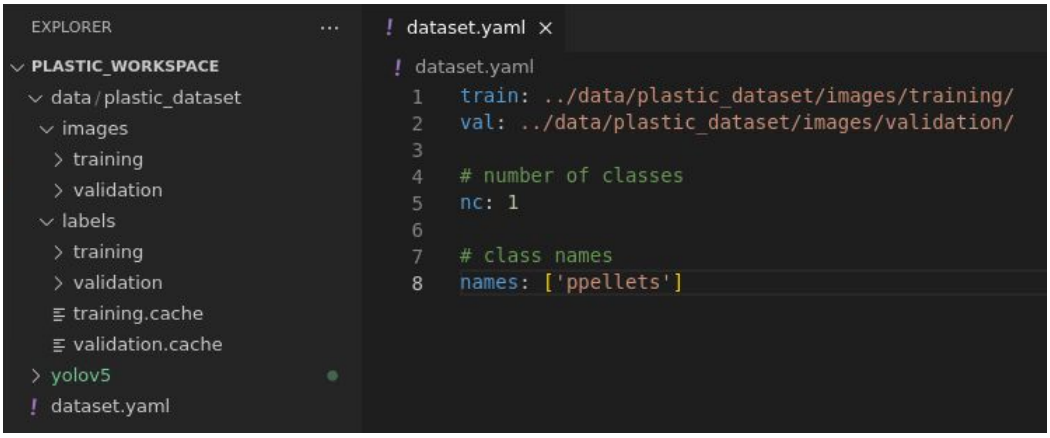


Figure 4.4: Necessary directory structure for the YOLOv5 implementation and contents of the dataset.yaml file.

The hyperparameters are left at their default settings and tuned further in the testing chapter. The train.py file is used for training YOLOv5 on the custom dataset. When running the train.py file in the terminal, parameters such as image size, batch size, epoch size, model architecture, dataset path, weights and more can be specified. Changing these parameters affects the detection capabilities of the YOLOv5 algorithm. Inference is done by using the detect.py file. The parameters specified when using the inference file are the source which is the path to the test images, weights which is the path to the weights of the custom trained model, conf which is objectness confidence, and name which saves the detections to a desired location. Listing 2 illustrates an example of using the train.py file and detect.py file.

```
!python train.py --img 640 --cfg yolov5s.yaml --batch 32 --epochs 100
--data dataset.yaml --weights yolov5s.pt --name plastic_pellets

!python detect.py --source ../dataset/images/test/
--weights runs/train/plastic_pellets/weights/best.pt --conf 0.25 --name plastic_pellets
```

Listing 2: Training and inference of the YOLOv5 algorithm

4.3 Sanity check

The result from the sanity check showed that the YOLOv5 algorithm trained on the beach environment dataset performs best when the small YOLOv5 architecture was chosen. Whereas, when trained with the medium and large architecture, the detection algorithm overfitted to the training data. The architecture did not matter as much for the YOLOv5 algorithm trained on the controlled dataset. Only the training time was impacted by the architecture choice. The large architecture achieved optimal training results in fewer epochs than the small and medium architecture. Additionally, the YOLOv5 algorithm did not have significant improvement after a certain amount of epochs, for the beach environment dataset it was after 50 epochs, whereas for the controlled dataset, it was 100 epochs.

From the sanity check it can be concluded that the most optimal choice for testing the detection algorithm on the controlled dataset, is to train it for 100 epochs with the large YOLOv5 architecture. The optimal choice for the beach environment dataset is 50 epochs with the small YOLOv5 architecture. These configurations will be used for testing in chapter 5.

Chapter 5

Testing

This chapter aims to evaluate the plastic pellet detection algorithm on different datasets. First, the evaluation metrics are explained. Next, the test approach is detailed. Afterward, the test results of the preliminary evaluation are given. Thereafter, the results of testing the beach environment dataset are detailed. At the end, the performance of the detection algorithm on each dataset is compared to the requirements.

5.1 Evaluation metrics

The metrics used to evaluate the performance of the detection algorithm are precision, recall, mean average precision (mAP), objectness loss, and box loss.

Intersection of union

The intersection of union (IoU)[86] describes the overlap between the predicted bounding box and the ground-truth bounding box. Whether a prediction is considered valid or not depends on the IoU. With an IoU of greater than 0.5, then the predicted bound box needs to have an overlap ratio of a minimum of 50%, with the area covered by the ground-truth bounding box and the predicted bounding box. in order to be considered a valid prediction. Predictions with an IoU score greater than the threshold are known as true positives, while predictions less than the threshold are known as false positives. Equation 5.1 shows how IoU is calculated.

$$\text{IoU} = \frac{\text{Area of Intersection}}{\text{Area of Union}} \tag{5.1}$$

Precision

The precision metric[86] is used to measure the accuracy of the model's predictions of true positives over all the predictions of the model. A true positive is a prediction of a class that belongs to that class, whereas a false positive is a prediction of a class that does not belong to that class. Whether a prediction is a true positive or false positive is determined by matching the ground-truth label with the predicted label, if they match then it is a true positive prediction, if no match then it is a false positive prediction. Equation 5.2 shows how the precision metric is calculated.

$$\text{Precision} = \frac{\text{True postives}}{\text{True postives + False postives}} \tag{5.2}$$

Recall

The recall metric[86] uses all true positive instances in the dataset, which means it also uses the true positive in the dataset that the model has not predicted, to determine a model's detecting ability of true positives. Equation 5.3 shows the calculation process of the recall, where false negatives are instances of a class belonging to that class but are not predicted to be part of that class.

$$\text{Recall} = \frac{\text{True postives}}{\text{True postives + False negatives}} \tag{5.3}$$

Trade off

A model with high precision but low recall has a small occurrence of false positives, meaning the model is good at predicting a class that belongs to that class. Low recall means that the model has many false negative instances, meaning the model misses some predictions that are an actual part of that class. In the case of a model with high recall and low precision, the model has a low occurrence of false negatives and a higher rate of false positives.

Mean average precision

The mAP[86] uses the precision and recall metrics to measure the performance of a detection algorithm. Equation 5.4 shows how the mAP metric is calculated, where it can be seen that the mAP metric is calculated by summing the average precision (AP) of each class (k) and dividing it by the number of classes (K). AP is calculated based on the precision-recall curve of each class. The precision-recall curve is plotted at a certain IoU. Figure 5.1 shows an illustration of the precision-recall curve.

$$\text{mAP} = \frac{1}{\text{K}} \sum_{k=1}^{k=K} AP_k \tag{5.4}$$

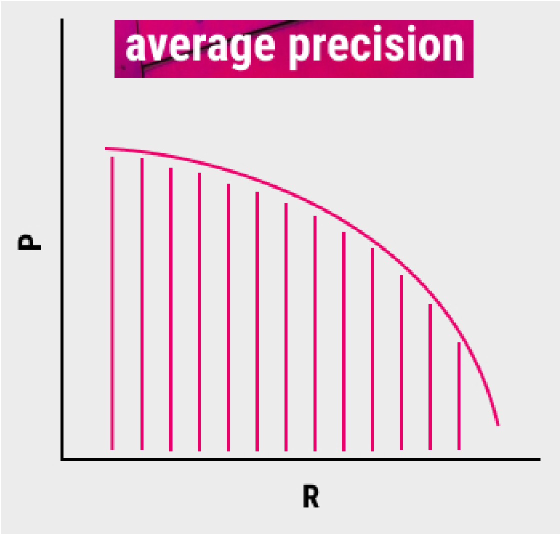


Figure 5.1: The image shows the precision-recall curve for calculating the average precision[87].

Loss

Loss functions are used to improve the detection capabilities of object detection algorithms. The performance of a model is improved by minimizing the loss function. YOLOv5 minimizes the objectness loss[88] and the box loss[88] in order to improve its detections. The objectness loss ensures that the regions with the highest probabilities of containing an object are chosen, which also helps in reducing the number of proposed anchor boxes, as the anchor boxes with high objectness loss are discarded. Box loss describes how good the coverage of a predicted bounding box is on an object, and helps the network predict bounding boxes that cover the objects better. The behavior of the loss functions follows a decreasing pattern up to a certain point, thereafter it no longer decreases but stays constant.

5.2 Tests

The detection algorithm is tested on two datasets. The first dataset is a laboratory setup, described in section 4.2. The images in the first dataset are on average captured 14cm from the ground. Tests with the first dataset serve as a proof of concept to validate whether plastic pellets can be detected in close-up (14cm in this case) images or not.

The second dataset is images of actual beach environments, the dataset is described in section 3.2.2, and the images are on average captured between 45cm-55cm from the ground. The second test explores how the performance of the detection algorithm is impacted when the images are from actual beach environments, that contain plastic debris, beach matter, and more. Additionally, the second test also serves as a baseline that can be further

improved on in future iterations of this project.
The first dataset consists of 100 annotated images and the second consists of 300 annotated images. Both of the datasets are randomly split into 80% training, 10% validation, and 10% testing.

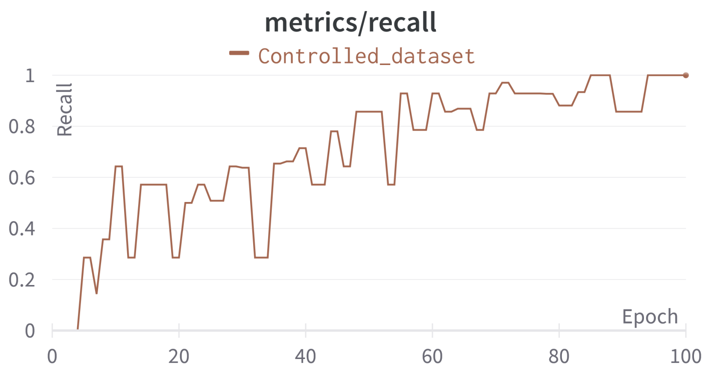
5.2.1 Preliminary evaluation

Before training the detection algorithm on a complex environment like the beach, it is trained on a controlled environment, where there is no plastic debris, no beach plant life, no beach animal life, and more. The controlled environment only contains sand, stones, and plastic pellets whose positions are known. The detection algorithm yielded the best result on the controlled dataset when trained for 100 epochs, with a batch size of four on the large YOLOv5 architecture, as described in section 4.3. Epoch 95 gave the most optimal results, which are shown in table 5.1. Figure 5.2 shows how the recall, precision, and mAP improve while the model is training. Figure 5.3 shows how the losses are improved in training. Table 5.2 shows the performance of the model on images that it has not seen before.

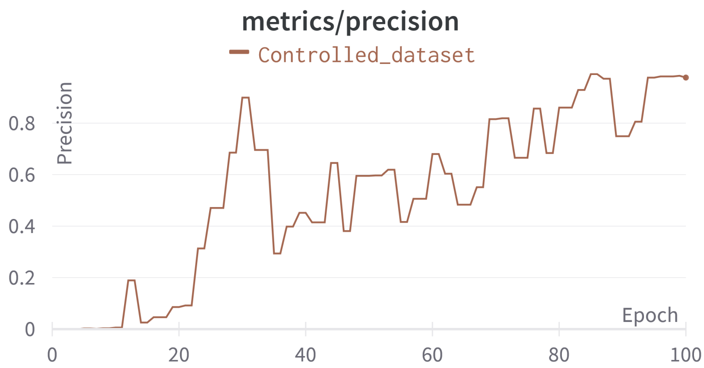
Training results

Evaluation of YOLOv5 on the controlled dataset.					
Controlled Dataset	Precision	Recall	mAP	objectness loss	Box loss
Model performance	0.977	1	0.995	-	-
Training	-	-	-	0.02765	0.03786
Validation	-	-	-	0.02148	0.03581

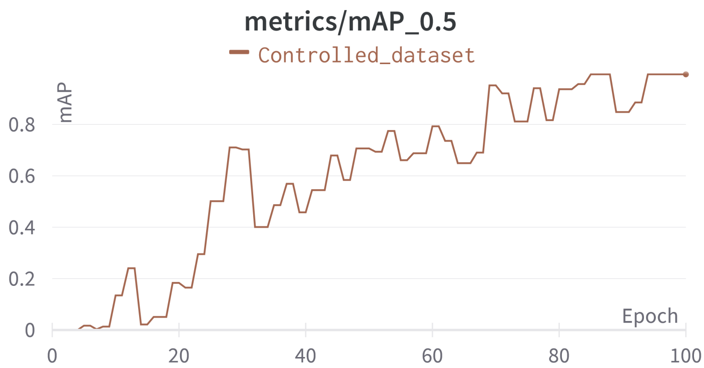
Table 5.1: Evaluation results of the YOLOv5 model on the controlled dataset.



(a) The recall score of YOLOv5 on the controlled dataset.

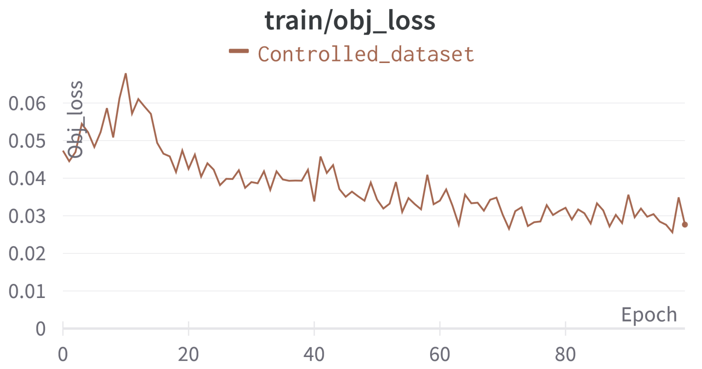


(b) The precision score of YOLOv5 on the controlled dataset.

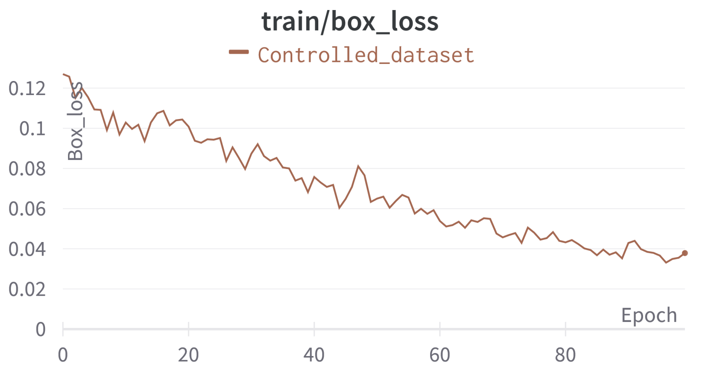


(c) The mAP score of YOLOv5 on the controlled dataset.

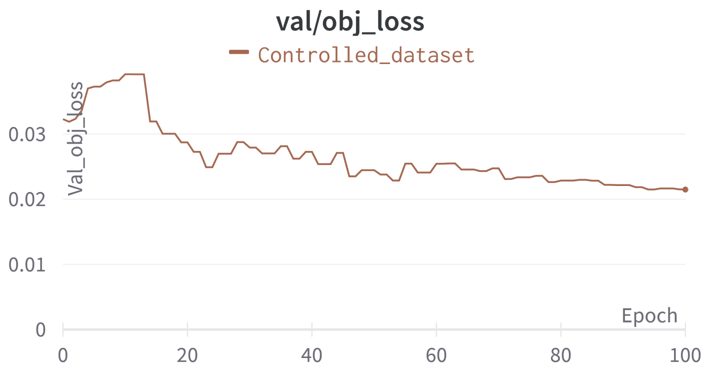
Figure 5.2: Overview of the three scores used to evaluate the performance of the detection algorithm on the controlled dataset.



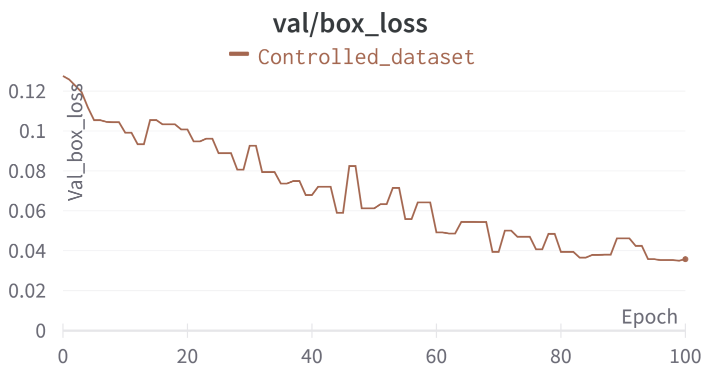
(a) Objectness loss of the training split.



(b) Box loss of the training split.



(c) Objectness loss of the validation split.



(d) Box loss of the validation split.

Figure 5.3: Overview of the four different losses observed while training on the controlled dataset.

Inference results

Inference results of test dataset.			
Test set of the controlled dataset	Precision	Recall	mAP
Inference performance	0.976	1	0.995

Table 5.2: Inference results of the YOLOv5 trained on the controlled dataset.

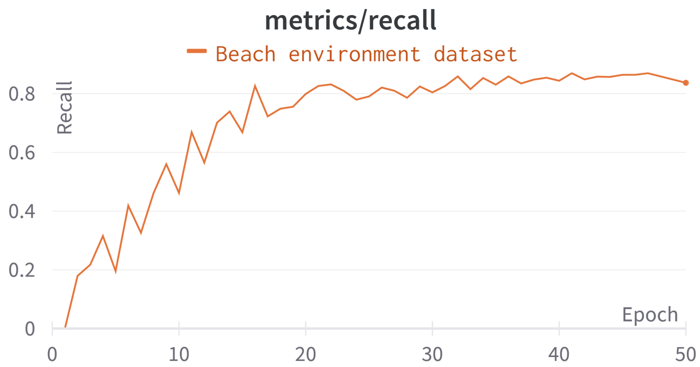
5.2.2 Beach environment dataset evaluation

The beach environment is not only complex due to the beach material and plastic material found there but also because of occlusion and reflection. Plastic pellets can be occluded by either the beach material or larger plastic material. Additionally, plastic material reflecting light from the sun can make recognition of plastic pellets difficult, as they will look saturated in the images. For the beach environment dataset, the detection algorithm gave optimal results when trained for 50 epochs, with a batch size of 16 on the small YOLOv5 architecture, as described in section 4.3. Epoch 37 gave the best results, which are shown in table 5.3. Figure 5.4 shows how the recall, precision, and mAP improve while the model is training. Figure 5.5 shows how the losses are improved in training. Table 5.4 shows the inference results of the trained model.

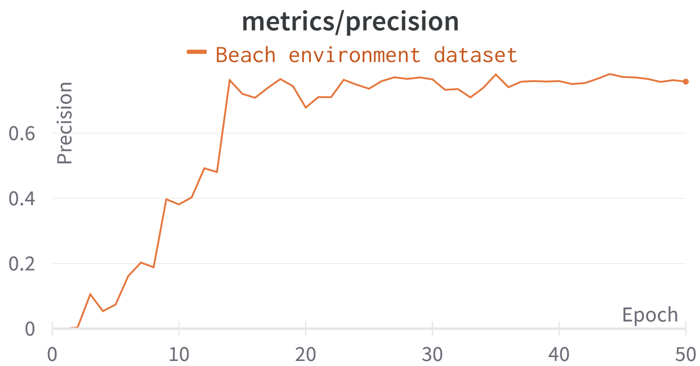
Training results

Evaluation of YOLOv5 on the actual beach environment dataset.					
Controlled Dataset	Precision	Recall	mAP	objectness loss	Box loss
Model performance	0.758	0.835	0.850	-	-
Training	-	-	-	0.02887	0.02868
Validation	-	-	-	0.01950	0.02088

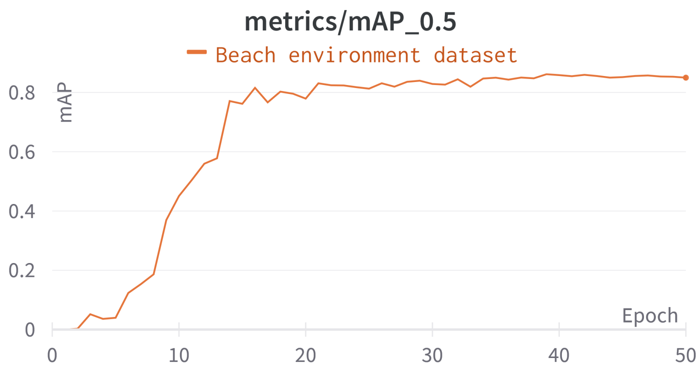
Table 5.3: Evaluation results of the YOLOv5 model on the actual beach environment dataset.



(a) The recall score of YOLOv5 on the actual beach environment dataset.

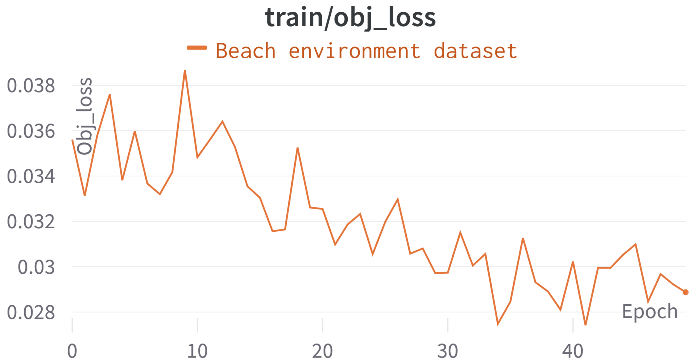


(b) The precision score of YOLOv5 on the actual beach environment dataset.

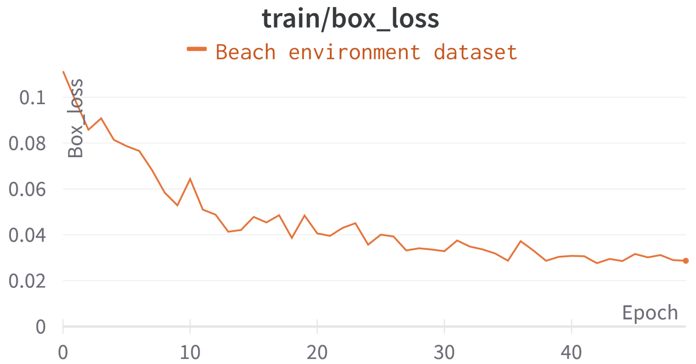


(c) The mAP score of YOLOv5 on the actual beach environment dataset.

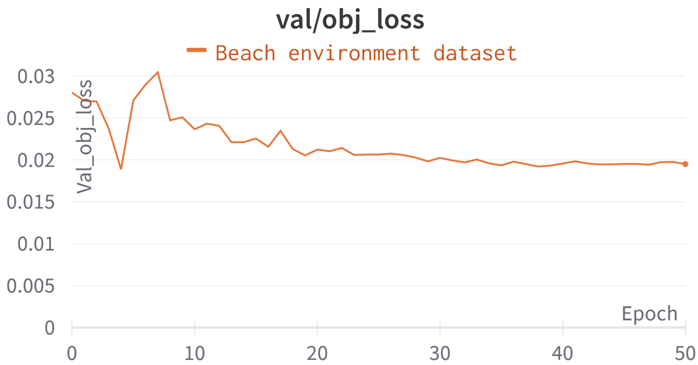
Figure 5.4: Overview of the three scores used to evaluate the performance of the detection algorithm on the actual beach environment dataset.



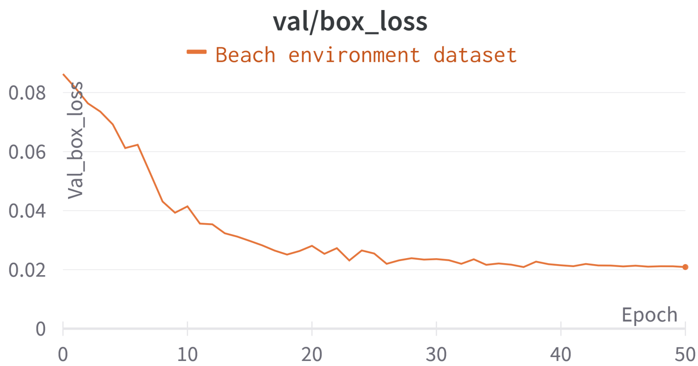
(a) Objectness loss of the training split.



(b) Box loss of the training split.



(c) Objectness loss of the validation split.



(d) Box loss of the validation split.

Figure 5.5: Overview of the four different losses observed while training on the actual beach environment dataset.

Inference results

Inference results of test dataset.			
Test set of the beach dataset	Precision	Recall	mAP
Inference performance	0.854	0.939	0.922

Table 5.4: Inference results of the YOLOv5 trained on the beach environment dataset.

5.2.3 Requirement checks

This section aims to confirm whether the detection algorithm has passed the requirements outlined in section 2.5.1 or not. The requirements are checked for the detection algorithm trained on the controlled dataset and the beach environment dataset separately. Additionally, the images from the training and inference are inspected to ensure that the requirements that cannot be measured are either passed or failed.

1. *An RGB camera must be used for data collection.*
   - The beach environment dataset passed.
   - The controlled dataset passed.
2. *Plastic pellets of different colors must be detectable.*
   - The beach environment dataset passed.
   - The controlled dataset passed.
3. *Plastic pellets that are translucent must be detectable.*
   - The beach environment dataset passed.
   - The controlled dataset passed.
4. *The detection algorithm must have a precision of at least 0.70.*
   - The beach environment dataset passed.
   - The controlled dataset passed.
5. *The detection algorithm must have a recall of at least 0.90.*
   - The beach environment dataset failed.
   - The controlled dataset passed.
6. *The detection algorithm must have a mAP of at least 0.85.*
   - The beach environment dataset passed.
   - The controlled dataset passed.

Chapter 6

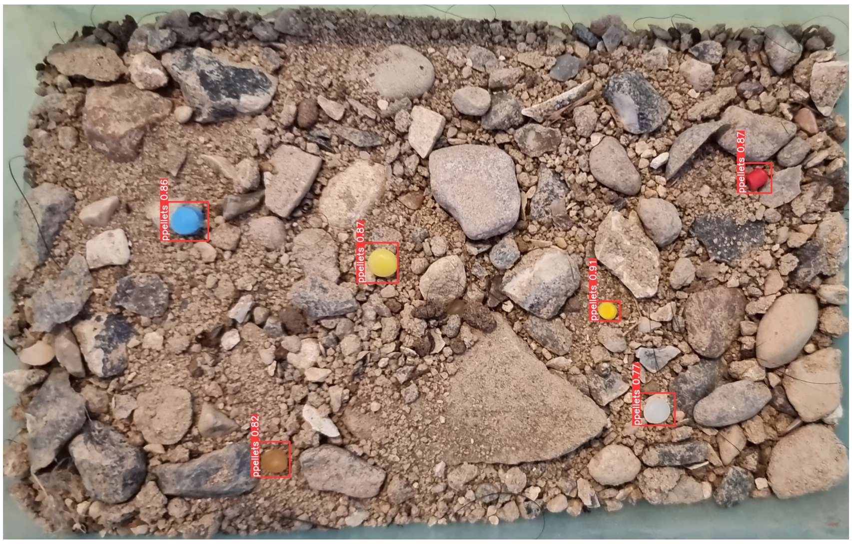
Discussion

6.1 Results of preliminary evaluation

The training results of the preliminary evaluation are shown in table 5.1. Comparing the requirements to the results in the table and inspecting the images used in training and validation, showed that requirements 1, 2, 3, 4, 5, and 6 are achieved. The scores reflect the detection algorithm's capabilities to detect plastic pellets as very accurate. The high performance is mainly due to the controlled dataset being in an ideal environment, that does not contain any negative factors that could impact the performance of the detector, whereas the actual beach environments have numerous factors that impact the detector's ability to spot plastic pellets, and an annotator's ability to label the dataset. The different negative factors are discussed further in section 6.3. The ideal image setup from the controlled dataset is shown in figure 6.1a, where plastic pellets of various types are placed on the sand surface. The location and shape information of the plastic pellets is taken into account when the dataset is annotated, which ensures that no plastic pellets are missed. Additionally, the images are captured close-up(14cm from the ground) to make sure the detector learns rich features of the plastic pellets and to make the labeling task less difficult.

From the inference results shown in table 5.2 and from manually inspecting the inferred images, confirmed that requirement 1, 2, 3, 4, 5, and 6 was achieved. The high inference results indicate that the model has achieved great plastic pellet recognition capabilities in its training phase. Figure 6.1b shows the results from inference, where it can be seen that the detector finds plastic pellets with the colors: yellow, blue, red, black, gray, white, brown, and translucent plastic pellets. Additionally, the detector is also capable of finding plastic pellets in different shades of the colors mentioned. The detector being capable of finding different color shades indicates that it could possibly also detect plastic pellets of entirely different colors than the ones it has trained on.

(a) Image of different types of plastic pellets placed in an ideal environment.



(b) Image of plastic pellets detected in an ideal environment. The image is from the controlled dataset. The detector is capable of finding all the different types of plastic pellets in the image.

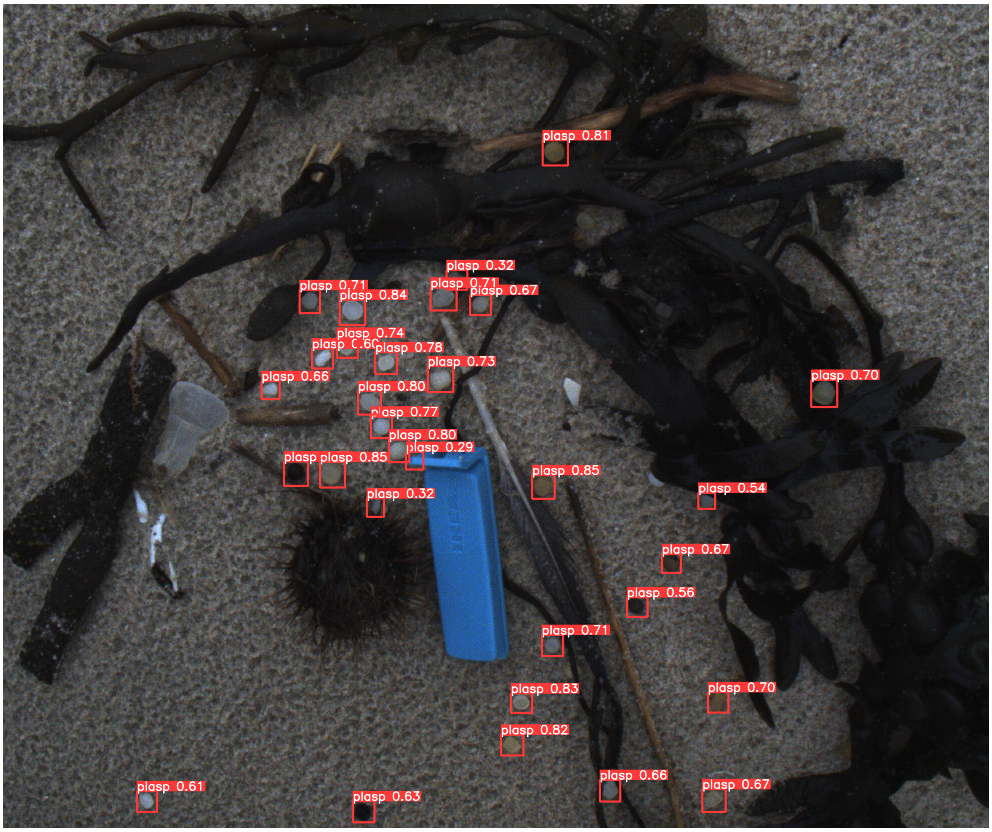
Figure 6.1

6.2 Results of Beach environment dataset evaluation

Table 5.3 shows training results from evaluating the detection algorithm on the actual beach environment dataset. The results from table 5.3, combined with a manual inspection of the images used for training and validation showed that requirement 1, 2, 3, 4 and 6 was achieved, while requirement 5 was not achieved. Even though the detector failed requirement 5, it is still quite accurate, as it only fell short by 0.07 on its recall score. The high performance of the detector is quite surprising when the state of the beach environment is taken into consideration. Figure 3.2 shows the different variants of beach environments and how polluted they are with plastic material, trash debris, beach material, and more. These factors increase the complexity of the dataset and the annotation task. Additionally, the images being captured 45cm-55cm on average from the ground to the camera also contribute to an increase in the annotation task. The difficulty in annotation lies in being able to spot the plastic pellets in the images. Figure 6.2 illustrates the difficulty, as it is very difficult to see where the plastic pellets are, especially the black plastic pellets. The trash debris makes it impossible to know whether the black objects are trash debris or plastic pellets. A dataset that is not annotated well enough, heavily impacts the performance of the object detection algorithm, as it is from the dataset it learns important features of the object it needs to recognize.

Figure 6.2: The image shows black plastic pellets mixed with black waste debris. It is not possible to determine what is what.

From the inference results shown in table 5.4 and by manually inspecting the inferred images, confirmed that requirement 1, 2, 3, 4, 5, and 6 was achieved. The high performance of the inference validates that the training phase of the model, on the beach environment dataset was a success, even though the training phase failed requirement 5. The manual inspection of the results from the inference showed that the inference results depend on the beach environment. For simple beach environments that are not polluted with trash and beach material, the detector manages to find almost every plastic pellet in the image. Images with greater complexity have occurrences of false positives and false negatives. Figure 6.4 shows inference detections where it can be seen that translucent, yellow, brown, black, and white plastic pellets and different shades of those colors are detected.



(a) Image of plastic pellets detected in a complex environment. The image is from the actual beach environment dataset. The image contains false positives and false. negatives detections.

(b) Image of plastic pellets detected in a simple environment. The image is from the actual beach environment dataset. The plastic pellet not found are not on the sand surface.

Figure 6.3: The images show the inference results from the actual beach environment dataset. The environment in 6.3a is quite complex when compared to the environment in 6.3b.

6.3 Challenges

When the solution for detecting plastic pellets was initially proposed in section 2.4 it was based on theory and intuition. At that point, it was unknown what actual challenges would arise throughout the process. Nevertheless, two important assumptions about the difficulties ahead were justified. The first assumption was that plastic debris and plastic pellets can be differentiated from one another based on their shapes and degradation factor. This assumption did not hold true for the dataset of the actual beach environment. In the dataset, the majority of the images which contained black plastic pellets and black trash debris, the black plastic pellets were impossible to differentiate from the black trash debris. Figure 6.4c illustrates the problem with black plastic pellets and black trash debris, where at the bottom of the figure a small area can be seen with either black plastic pellets or black trash debris. It is not possible to determine which it is even if zoomed in at pixel level, figure 6.2 shows the same problem.

The second assumption is the core of the entire solution, and it also influences the first assumption. The second assumption was formulated as *close-up RGB images of plastic pellets reduce the difficulty in finding plastic pellets and increase an object detection algorithm's recognition ability.* This assumption held true for both datasets. The laboratory dataset especially proved the assumption by achieving almost perfect performance values. Where the main contributors to the almost perfect performance are the ideal environment and the close-up(14cm from the ground) images of the plastic pellets. Whereas the images from the actual beach environment dataset were captured 45cm-55cm on average from the ground-to-camera which is about three times as far as the controlled dataset. The beach environment's ground-to-camera distance is part of the problem in assumption one when trying to differentiate between plastic pellets and trash debris. With a lower ground-to-camera distance, the different objects in the environment would be much less difficult to differentiate between, which is illustrated in figure 6.4. Figure 6.4 shows three images captured at three different heights, the ground-to-camera distance in 6.4a is 5cm, 6.4b 20cm, and 6.4c 45cm-55cm. The distance used in 6.4a is ideal for complex environments that are polluted with trash debris, as the low ground-to-camera distance ensures a clear view of the environment and the objects in it. The plastic pellet seen in 6.4a becomes difficult to see in 6.4b, where the ground-to-camera distance is four times larger. Without having prior knowledge of the position of the plastic pellet, the image would need to be zoomed in on and scanned for the object. With the ground-to-camera distance in 6.4c the difficulty of finding plastic pellets becomes very high.

(a) An image from the controlled dataset, where the ground-to-camera distance is 5cm. The plastic pellet is very easy to find. The plastic pellet is almost in the center of the image.

(b) An image from the controlled dataset, where the ground-to-camera distance is 20cm. The plastic pellet is more difficult to find than in 6.4a. The plastic pellet is in the top right corner.

(c) An image from the actual beach environment dataset, where the ground-to-camera distance is 45cm-55cm. It is not possible to determine the objects in the image due to high the ground-to-camera distance

Figure 6.4: The figure shows three images from three different ground-to-camera distances and finding the objects in each increase in difficulty, from 6.4a towards 6.4c.

Having prior knowledge of the position and shapes of the plastic pellets also impacts the annotating difficulty. For the annotation of the controlled dataset, prior knowledge was used, which was possible because the dataset was made by the person annotating it. Whereas the actual beach environment dataset was given by the Race to Ocean Foundation, thus no prior knowledge could be used for the annotation task, hence the difficulty of the task increased.

From the results it is concluded that the three main factors that have impacted the performance of the detection algorithm are the ground-to-camera distance, debris materials, and prior knowledge of the dataset.

6.4 Future work

The time limit leaves future work in optimization and additional implementation of the project.

The dataset acquired from the Race for Ocean Foundation has difficulties that are discussed in section 6.3. If the difficulties with the dataset are solved, then the detection algorithm's performance would almost reach perfection, similar to the controlled dataset. One way to go about it would be to create a new dataset of the actual beach environments, where the annotation is done right after the images are captured, which would ensure high labeling quality.

A more complex addition to the plastic pellet detector would be to expand on the detection. Instead of detecting plastic pellets only, the detector could learn features from different trash material or debris material that is found on the beach, hence the detector would be able to detect more waste that is harmful to the beaches and coastal environments. It would also allow a robot solution to use a complete detection system for removing the beaches and coastal areas for all sorta waste.

Chapter 7

Conclusion

The goal of this project was to investigate possible solutions, for the detection of plastic pellets on the beach sand surface, from the research choose an optimal solution for development, and ensure that the developed solution functions in actual beach environments.

To better understand the severity of plastic pollution, the problem analysis investigated areas where plastic is used, the evolution of plastic production, plastic's disposability and recyclability properties, manufacturing of plastic, and how plastic pellets leak into the environment. The investigation showed that plastic is widely used in almost all fields, and the production of plastic has increased by 70% (190 million tonnes) in less than a decade(2010-2019). The rapid growth has caused plastic material to massively leak into the environment and damage it. Furthermore, with plastic's poor disposability and recyclability properties and plastic being primarily manufactured from crude oil and natural gas, the damage to the environment becomes very long-termed and severe. A plastic material that was identified as difficult to see and clean from the environment, was plastic pellets. Beach and coastal areas were identified as the place where the plastic pellets would wash up and pollute the area. After further research on how plastic pellets could be detected, a final problem definition based on the knowledge from the research was formulated as such:

Can images of beach sand surfaces from an RGB camera be used to train a YOLOv5 algorithm, for an autonomous detection system of solely plastic pellets on the beach sand surface?

The plastic pellet detector showed itself to be capable of detecting plastic pellets in various beach sand surface environments. On images from a controlled environment, the detector's best score was a precision of 0.977, a recall of 1, and an mAP of 0.995. On images from actual beach environments, the detector's best score was a precision of 0.758, a recall of 0.835, and an mAP of 0.850. Based on the results, it can be concluded that RGB images of plastic pellets in a beach sand surface environment can be used by a detection algorithm to recognize the plastic pellets.

With future iterations of this project, additional waste classes could be added for the detector to learn to recognize, in order to remove multiple waste materials at once from the beaches and coastal areas. Additionally, the performance on the actual beach environment dataset of the plastic pellet detector could be improved even further by keeping track of the plastic pellets in the images or by doing annotation on the fly.

Bibliography

[1] Shashi Bahl et al. "Biodegradation of plastics: A state of the art review". In: *Materials Today: Proceedings* 39 (2021). International Conference on Advanced Materials and Modern Manufacturing, pp. 31–34. ISSN: 2214-7853. DOI: https://doi.org/10.1016/j.matpr.2020.06.096. URL: https://www.sciencedirect.com/science/article/pii/S2214785320345508.

[2] Sarah C Gall and Richard C Thompson. "The impact of debris on marine life". In: *Marine pollution bulletin* 92.1-2 (2015), pp. 170–179.

[3] *Clearbot*. https://www.clearbot.org/.

[4] *Wasteshark*. https://www.ranmarine.io/.

[5] *Mr. Trash Wheel*. https://www.mrtrashwheel.com/.

[6] *The ocean clean up interceptor*. https://theoceancleanup.com/rivers/.

[7] *Bebot*. https://searial-cleaners.com/our-cleaners/bebot-the-beach-cleaner/.

[8] *Plastics Applications*. https://www.bpf.co.uk/plastipedia/applications/Default.aspx.

[9] Hannah Ritchie and Max Roser. "Plastic Pollution". In: *Our World in Data* (2018). https://ourworldindata.org/plastic-pollution.

[10] Matthew MacLeod et al. "The global threat from plastic pollution". In: *Science* 373.6550 (2021), pp. 61–65.

[11] Hannah Ritchie and Max Roser. "Plastic pollution". In: *Our World in Data* (2018).

[12] Penghui Li et al. "Characteristics of plastic pollution in the environment: a review". In: *Bulletin of environmental contamination and toxicology* 107 (2021), pp. 577–584.

[13] Gajahin Gamage Nadeeka Thushari and Jayan Duminda Mahesh Senevirathna. "Plastic pollution in the marine environment". In: *Heliyon* 6.8 (2020), e04709.

[14] Christopher J Rhodes. "Plastic pollution and potential solutions". In: *Science progress* 101.3 (2018), pp. 207–260.

[15] Md Imran Hossain and Md Akber Subahan Mahbub Tuha. "Biodegradable plastic production from daily Household waste materials and Comparison the decomposing time with synthetic polyethylene plastic". In: *International Journal of Advancement in Life Sciences Research* 3.3 (2020), pp. 16–19.

[16] *PlasticsEurope Operation Clean Sweep*. https://www.opcleansweep.eu/application/files/8316/3456/6233/PlasticsEurope_OCS_progress_report-2017.pdf.

[17] Dipak Kumar Shaw and Pranav Sahni. "Plastic to oil". In: *Journal of Mechanical and Civil Engineering* (2014), pp. 46–48.

[18] *common types of plastic*. https://clearonplastics.com/what-are-the-types-of-plastic-and-what-are-they-used-for/.

[19] *Containers Lost at Sea*. https://static1.squarespace.com/static/5ff6c5336c885a268148bdcc/t/62b2d93516eba2026ec97cee/1655888181926/Containers_Lost_at_Sea_2022.pdf.

[20] Jenna R. Jambeck et al. "Plastic waste inputs from land into the ocean". In: *Science* 347.6223 (2015), pp. 768–771. DOI: 10.1126/science.1260352. eprint: https://www.science.org/doi/pdf/10.1126/science.1260352. URL: https://www.science.org/doi/abs/10.1126/science.1260352.

[21] Xiangtao Jiang et al. "Occurrence, distribution, and associated pollutants of plastic pellets (nurdles) in coastal areas of South Texas". In: *Science of The Total Environment* 842 (2022), p. 156826.

[22] Hrissi K Karapanagioti and Irene Klontza. "Testing phenanthrene distribution properties of virgin plastic pellets and plastic eroded pellets found on Lesvos island beaches (Greece)". In: *Marine environmental research* 65.4 (2008), pp. 283–290.

[23] Daniel Gorman et al. "Organic contamination of beached plastic pellets in the South Atlantic: Risk assessments can benefit by considering spatial gradients". In: *Chemosphere* 223 (2019), pp. 608–615.

[24] HB Jayasiri, CS Purushothaman, and A Vennila. "Bimonthly variability of persistent organochlorines in plastic pellets from four beaches in Mumbai coast, India". In: *Environmental monitoring and assessment* 187 (2015), pp. 1–13.

[25] Murray R Gregory. "Plastic pellets on New Zealand beaches". In: *Marine Pollution Bulletin* 8.4 (1977), pp. 82–84.

[26] JG Shiber. "Plastic pellets on the coast of Lebanon". In: *Marine Pollution Bulletin* 10.1 (1979), pp. 28–30.

[27] Shiye Zhao, Lixin Zhu, and Daoji Li. "Characterization of small plastic debris on tourism beaches around the South China Sea". In: *Regional Studies in Marine Science* 1 (2015), pp. 55–62.

[28] Hamed Masoumi, Seyed Mohsen Safavi, and Zahra Khani. "Identification and classification of plastic resins using near infrared reflectance". In: *Int. J. Mech. Ind. Eng* 6 (2012), pp. 213–220.

[29] Stas Zinchik et al. "Accurate Characterization of Mixed Plastic Waste Using Machine Learning and Fast Infrared Spectroscopy". In: *ACS Sustainable Chemistry & Engineering* 9.42 (2021), pp. 14143–14151.

[30] UK Adarsh et al. "Spectroscopy: A promising tool for plastic waste management". In: *TrAC Trends in Analytical Chemistry* (2022), p. 116534.

[31] Abraham Vázquez-Guardado et al. "Multi-spectral infrared spectroscopy for robust plastic identification". In: *Applied optics* 54.24 (2015), pp. 7396–7405.

[32] Shichao Zhu et al. "Plastic solid waste identification system based on near infrared spectroscopy in combination with support vector machine". In: *Advanced Industrial and Engineering Polymer Research* 2.2 (2019), pp. 77–81.

[33] Jennifer Cocking et al. "Aerial detection of beached marine plastic using a novel, hyperspectral short-wave infrared (SWIR) camera". In: *ICES Journal of Marine Science* 79.3 (Feb. 2022), pp. 648–660. ISSN: 1054-3139. DOI: 10.1093/icesjms/fsac006. eprint: https://academic.oup.com/icesjms/article-pdf/79/3/648/43513594/fsac006.pdf. URL: https://doi.org/10.1093/icesjms/fsac006.

[34] "Deep learning for chemometric analysis of plastic spectral data from infrared and Raman databases". In: *Resources, Conservation and Recycling* 188 (2023), p. 106718. ISSN: 0921-3449. DOI: https://doi.org/10.1016/j.resconrec.2022.106718. URL: https://www.sciencedirect.com/science/article/pii/S0921344922005511.

[35] E. Maris et al. "Polymer tracer detection systems with UV fluorescence spectrometry to improve product recyclability". In: *Minerals Engineering* 29 (2012). Sustainability through Resource Conservation and Recycling, pp. 77–88. ISSN: 0892-6875. DOI: https://doi.org/10.1016/j.mineng.2011.09.016. URL: https://www.sciencedirect.com/science/article/pii/S0892687511003633.

[36] Juraj Jasik et al. "Vacuum ultraviolet laser-induced breakdown spectroscopy analysis of polymers". In: *Spectrochimica Acta Part B: Atomic Spectroscopy* 64.10 (2009). A Collection of Papers Presented at the Fifth International Conference on Laser Induced Breakdown Spectroscopy (LIBS 2008), pp. 1128–1134. ISSN: 0584-8547. DOI: https://doi.org/10.1016/j.sab.2009.07.013. URL: https://www.sciencedirect.com/science/article/pii/S0584854709002250.

[37] S Gies et al. "Investigating the Potential of UV-excited Photoluminescence Spectroscopy for the Identification of Plastics". In: *CEST 2019 Conference, Rhodos*. 2019.

[38] Janusz Bobulski and Mariusz Kubanek. "Deep learning for plastic waste classification system". In: *Applied Computational Intelligence and Soft Computing* 2021 (2021), pp. 1–7.

[39] Jaeseok Kim et al. "An Innovative Automated Robotic System based on Deep Learning Approach for Recycling Objects." In: *ICINCO (2)*. 2019, pp. 613–622.

[40] Bing Xue et al. "Deep-sea debris identification using deep convolutional neural networks". In: *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing* 14 (2021), pp. 8909–8921.

[41] Dillam Jossue Diaz Romero et al. "Enhanced Plastic Recycling using RGB+ Depth Fusion with MassFaster and MassMask R-CNN". In: *2022 Fourth International Conference on Transdisciplinary AI (TransAI)*. IEEE. 2022, pp. 22–29.

[42] Muhammad Faisal et al. "Faster R-CNN Algorithm for Detection of Plastic Garbage in the Ocean: A Case for Turtle Preservation". In: *Mathematical Problems in Engineering* 2022 (2022).

[43] Mattis Wolf et al. "Machine learning for aquatic plastic litter detection, classification and quantification (APLASTIC-Q)". In: *Environmental Research Letters* 15.11 (2020), p. 114042.

[44] Zhenxing Cai et al. "Research on Waste Plastics Classification Method Based on Multi-Scale Feature Fusion". In: *Sensors* 22.20 (2022), p. 7974.

[45] Paul Viola and Michael J Jones. "Robust real-time face detection". In: *International journal of computer vision* 57.2 (2004), pp. 137–154.

[46] Navneet Dalal and Bill Triggs. "Histograms of oriented gradients for human detection". In: *2005 IEEE computer society conference on computer vision and pattern recognition (CVPR'05)*. Vol. 1. Ieee. 2005, pp. 886–893.

[47] Pedro Felzenszwalb, David McAllester, and Deva Ramanan. "A discriminatively trained, multiscale, deformable part model". In: *2008 IEEE conference on computer vision and pattern recognition*. Ieee. 2008, pp. 1–8.

[48] Shaoqing Ren et al. "Faster r-cnn: Towards real-time object detection with region proposal networks". In: *Advances in neural information processing systems* 28 (2015).

[49] Wei Liu et al. "Ssd: Single shot multibox detector". In: *European conference on computer vision*. Springer. 2016, pp. 21–37.

[50] Joseph Redmon et al. "You only look once: Unified, real-time object detection". In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2016, pp. 779–788.

[51] Manuel Carranza-García et al. "On the Performance of One-Stage and Two-Stage Object Detectors in Autonomous Vehicles Using Camera Data". In: *Remote Sensing* 13.1 (2021). ISSN: 2072-4292. DOI: `10.3390/rs13010089`. URL: `https://www.mdpi.com/2072-4292/13/1/89`.

[52] *[DeepReader] SSD: Single Shot MultiBox Detector*. `https://www.youtube.com/watch?v=BRPdiuWO6sA`.

[53] *Race for Ocean field trip*. https://www.youtube.com/watch?v=MJiOBwFy-aQ.

[54] *Race for Ocean about*. https://raceforoceans.org/about.

[55] *YOLO bounding box*. https://albumentations.ai/docs/getting_started/bounding_boxes_augmentation/.

[56] Iqbal H Sarker. "Machine learning: Algorithms, real-world applications and research directions". In: *SN computer science* 2.3 (2021), p. 160.

[57] *CNN classic architecture*. https://medium.com/@RaghavPrabhu/understanding-of-convolutional-neural-network-cnn-deep-learning-99760835f148.

[58] Keiron O'Shea and Ryan Nash. "An introduction to convolutional neural networks". In: *arXiv preprint arXiv:1511.08458* (2015).

[59] *Convolution operation*. https://learnopencv.com/understanding-convolutional-neural-networks-cnn/.

[60] Nils Bjorck et al. "Understanding batch normalization". In: *Advances in neural information processing systems* 31 (2018).

[61] *activation functions picture*. https://www.cs.cmu.edu/~bhiksha/courses/deeplearning/Spring.2019/archive-f19/www-bak11-22-2019/document/note/hwnotes/HW1p1.html.backup.

[62] *Pooling picture*. https://epynn.net/Pooling.html.

[63] Xu Zhang, Svebor Karaman, and S. Chang. *Detecting and Simulating Artifacts in GAN Fake Images*. July 2019.

[64] Seyed Iman Mirzadeh et al. "Understanding the role of training regimes in continual learning". In: *Advances in Neural Information Processing Systems* 33 (2020), pp. 7308–7320.

[65] Shoaib Siddiqui et al. "Automatic fish species classification in underwater videos: Exploiting pretrained deep neural network models to compensate for limited labelled data". In: *ICES Journal of Marine Science* 75 (May 2017). DOI: 10.1093/icesjms/fsx109.

[66] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. "Imagenet classification with deep convolutional neural networks". In: *Communications of the ACM* 60.6 (2017), pp. 84–90.

[67] Yann LeCun et al. "Gradient-based learning applied to document recognition". In: *Proceedings of the IEEE* 86.11 (1998), pp. 2278–2324.

[68] Karen Simonyan and Andrew Zisserman. "Very deep convolutional networks for large-scale image recognition". In: *arXiv preprint arXiv:1409.1556* (2014).

[69] Christian Szegedy et al. "Going deeper with convolutions". In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2015, pp. 1–9.

[70] Matthew D Zeiler and Rob Fergus. "Visualizing and understanding convolutional networks". In: *Computer Vision–ECCV 2014: 13th European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part I 13*. Springer. 2014, pp. 818–833.

[71] Kaiming He et al. "Deep residual learning for image recognition". In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2016, pp. 770–778.

[72] Simone Bianco et al. "Benchmark analysis of representative deep neural network architectures". In: *IEEE access* 6 (2018), pp. 64270–64277.

[73] Hongchao Zhuang et al. "High Inclusiveness and Accuracy Motion Blur Real-Time Gesture Recognition Based on YOLOv4 Model Combined Attention Mechanism and DeblurGanv2". In: *Applied Sciences* 11.21 (2021). ISSN: 2076-3417. DOI: 10.3390/app11219982. URL: https://www.mdpi.com/2076-3417/11/21/9982.

[74] Joseph Redmon and Ali Farhadi. "Yolov3: An incremental improvement". In: *arXiv preprint arXiv:1804.02767* (2018).

[75] Chien-Yao Wang et al. "CSPNet: A new backbone that can enhance learning capability of CNN". In: *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition workshops*. 2020, pp. 390–391.

[76] *BottleNeck*. https://github.com/ultralytics/yolov5/issues/6998#issue-1170533269.

[77] Shu Liu et al. "Path aggregation network for instance segmentation". In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2018, pp. 8759–8768.

[78] Eunchan Kim, Byungjoon Yoo, and Yeunwoong Kyung. "SHOMY: Detection of Small Hazardous Objects using the You Only Look Once Algorithm". In: 16 (Sept. 2022), pp. 2688–2703. DOI: 10.3837/tiis.2022.08.012.

[79] Kaiming He et al. "Spatial pyramid pooling in deep convolutional networks for visual recognition". In: *IEEE transactions on pattern analysis and machine intelligence* 37.9 (2015), pp. 1904–1916.

[80] *"Our SPP module is exactly the same"*. https://github.com/ultralytics/yolov5/issues/8785#issuecomment-1199931106.

[81] *Complete overview*. https://github.com/ultralytics/yolov5/issues/280#issuecomment-1001850116.

[82] *Samsung Galaxy S21 Ultra 5G camera specification*. https://www.samsung.com/nz/smartphones/galaxy-s21-ultra-5g/specs/.

[83] *Computer Vision Annotation Tool*. https://github.com/opencv/cvat.

[84] *PyTorch installation*. https://pytorch.org/get-started/locally/.

[85] *YOLOv5 github*. https://github.com/ultralytics/yolov5.

[86] Abanoub Ghobrial et al. "Evaluation Metrics for CNNs Compression". In: *arXiv preprint arXiv:2305.10616* (2023).

[87] *Average precision graph*. https://labelyourdata.com/articles/mean-average-precision-map.

[88] Margrit Kasper-Eulaers et al. "Short Communication: Detecting Heavy Goods Vehicles in Rest Areas in Winter Conditions Using YOLOv5". In: *Algorithms* 14.4 (2021). ISSN: 1999-4893. DOI: 10.3390/a14040114. URL: https://www.mdpi.com/1999-4893/14/4/114.