The development of an autonomous robot that detects and removes plastic on beaches

Student: Ramon Dijkstra — Supervisor: dr. Ilaria Tiddi

Vrije Universiteit, Amsterdam
July 2, 2020

Abstract

This paper examines to what extent Artificial Intelligence supports the development of an autonomous robot that detects and removes plastic on beaches. Plastic pollution is a worldwide problem: next to oceans and rivers, beaches are polluted with an extensive amount of plastic debris. There is currently no general method to remove all plastic on beaches. Besides that, manually removing this plastic is time-consuming. Our idea is to develop an autonomous robot that detects and removes plastic on beaches. This research does not focus on the mechanical side of the robot. It examines the extent to which path planning, object detection, and a knowledge graph are suitable technologies for the development of an autonomous beach cleaning robot. A simple area coverage path planning algorithm can provide the initial movement of the robot. The capabilities of an object detection program called YOLO are explored and it is found that it can detect objects with high precision and in real-time. A knowledge graph can represent the beach environment and the actions that need to be taken after an event occurs. The paper concludes that path planning, object detection, and a knowledge graph are suitable technologies for the development of an autonomous robot that detects and removes plastic on beaches.

Keywords — Autonomous Robot, Plastic Pollution,, Artificial Intelligence, Path Planning, Object Detection, Knowledge Graph

1 Introduction

This paper examines to what extent Artificial Intelligence supports the development of an autonomous robot that detects and removes plastic on beaches. The goal of this research is to set the first step towards the worldwide use of autonomous beach cleaning robots. This initiative is taken because plastic pollution is a massive problem that affects our whole planet [1, 2]. Large organizations like the European Commission[1], WWF[2], IUCN[3], and Greenpeace[4] all published reports on the problem of plastic and described possible solutions.

[1]https://eur-lex.europa.eu/legal-content/EN/TXT/?uri=COM%3A2018%3A28%3AFIN
[2]https://d2ouvy59p0dg6k.cloudfront.net/downloads/plastic_update_last_03_25.pdf
[3]https://portals.iucn.org/library/sites/library/files/documents/2014-067.pdf
[4]www.greenpeace.to/greenpeace/wp-content/uploads/2011/05/plastic_ocean_report.pdf

Oceans, rivers, and beaches are the places where the majority of the plastic debris is located [3–7]. Initiatives to remove the plastic debris are taken by organizations like The Ocean Cleanup[5]. Such organizations develop solutions from simply removing the plastic with a large group of people to more advanced technological solutions. The latter is our field of research.

This paper will focus on plastic debris on beaches. Oceans and rivers are thus related to the problem but not within our scope of research. At the moment, there is no general method to remove all the plastic from beaches. Several methods like removing the plastic through human group activities exist. However, humans are limited in their time and capabilities to clean beaches. Robots, on the other hand, are being employed for many tasks nowadays. They are being used in manufacturing, health care, agriculture, military, and others [8–11]. Robots in these sectors can complete tasks faster and more precise in comparison to humans. To take it one step further, autonomous robots can perform tasks without human interaction. The idea presented in this paper is an autonomous robot that detects and removes plastic on beaches. It is mechanically possible to develop a robot that could drive around the beach [12,13]. However, in this research, the software is taken into account. This paper suggests the idea of a software side based on AI techniques. The robot consists of three main elements that together contribute to the autonomy of the robot: namely, path planning, object detection, and a knowledge graph. The idea is that when a robot knows the path that needs to be walked, object detection can be used to identify objects, and a knowledge graph can be used to represent the environment and infer the actions that need to be taken when an object is detected.

Path planning is the task in robotics to find a path between two places which the robot is then able to walk. The initial movement of the robot is decided by a coverage path planning algorithm. The path should cover the area that needs to be cleaned [14]. A relatively simple algorithm would suit the job. Object detection is the task in computer science to identify objects based on images or videos [15]. Studies on object detection including YOLO and Fast R-CNN are taken into account [16–18]. Our object detection part has to be able to identify objects in real-time and with high precision. This object detection program is connected to a knowledge graph. Following the definition of Ehrlinger and Wöß, 'a knowledge graph acquires and integrates information into an ontology and applies a reasoner to derive new knowledge' [19]. Robobrain [20] and KnowRob [21] are both knowledge graphs created for autonomous robots. However, these are focused on creating a general knowledge graph for autonomous robots. Besides that, these knowledge graphs are created for an indoor environment. In contrast, this research is focused on the representation of a specific field, namely the outdoor beach environment. The autonomous robot will detect objects and will infer what actions it has to take based on the knowledge graph using SPARQL queries. This is the decision making part. The primary goal of this study is to answer the question to what extent path planning, object detection, and a knowledge graph support the development of an autonomous robot that detects and removes plastic on beaches.

[5]https://theoceancleanup.com/

2 Related Work

2.1 The problem of plastic pollution

Plastic pollution is a worldwide problem [1, 2]. Several studies showed that a high percentage of plastic waste is located on beaches all around the world [3–7]. Besides these studies, the European Commission, WWF, IUCN, and Greenpeace have all written reports about the impact of plastic debris and the actions that need to be taken. Thus, experts, nations, and large organizations all around the world are concerned with the amount of plastic debris that lands on beaches. To solve this, the current plastic should be removed and the flow of new plastic must be controlled [3]. However, it is difficult to remove the plastic and it is even more difficult to keep the flow of new plastic controlled. Several technological solutions are developed to deal with the removal of plastic [22, 23]. Also, organizations like The Ocean Cleanup are looking towards solutions to the problem of cleaning beaches. The main complication is that it is time-consuming to manually remove all plastic on beaches. Another complication is that it could be dangerous to remove large amounts of sand because that would damage the structure of beaches [24]. Keeping the flow of new plastic controlled is even more difficult because plastic is still widely used and an alternative to plastic is not easily found. There are thus two main problems. The first problem that is identified is that the removal of the current plastic debris is not easily accomplished by humans. The second is the difficulty of preventing plastic litter. This research will focus on solving the first problem.

2.2 Path planning

Path planning is the task in robotics to find a path between two points which can then be walked by the robot. Following the three principles of path planning, the path that is determined must be the shortest path, the determination of the path should be fast and the path should be adaptable [25]. Several algorithms for task planning were discovered [26]. Also, several robot navigation methods were looked into [27]. An example of a robot navigation method is laser navigation. The laser can identify the position of the robot. It then determines the next steps of the path that needs to be walked. The robot however does need to have a path in mind to determine these next steps. Thus, we first need to look into a path planning algorithm before this navigation method can be used. We also looked into path planning algorithms for autonomous robots [28]. Lastly, coverage path planning algorithms are taken into account [14]. We concluded that task planning, navigation, and coverage path planning algorithms are all suitable ways to perform path planning. For our scope, cleaning the beach can be seen as a coverage path planning problem. This is because the eventual goal of the creation of the autonomous robot is to clean the whole beach. The other techniques could potentially be used as our path planning algorithm but coverage path planning seems to suit our problem better. Besides the fact that the goal is to cover the whole beach area, the other techniques are more difficult to implement. Therefore, our research will focus on a coverage path planning algorithm. Within the scope of coverage path planning algorithms, the boustrophedon path stood out of the rest [29]. This is a simple path coverage algorithm. It is the fastest algorithm that covers an entire area.

2.3 Object detection

Object detection is the task in computer vision where computers assign objects a specific class from an image or video [15]. It is easy for humans to see the difference between a dog and a human. However, this can be very hard for a computer. This is because humans have background knowledge which can be seen as experience. Because of our experience, we can distinguish between different objects. A computer does not have this background knowledge but this can be achieved with machine learning. Deep learning is a subset of machine learning. It is based on artificial neural networks and it is similar to how humans learn from experience. An object detection algorithm is trained on a large number of images using deep learning [30]. It is then able to assign a class to a new image based on the experience that it obtained during training. We have studied several object detection systems [16]. YOLO and Fast R-CNN stood out of the rest because they can perform in real-time and with high precision [16]. YOLO is focused on real-time object detection and is ideal for fast real-time systems [17]. Fast R-CNN is similar to YOLO but YOLO is often preferred because it suits the real-life situation more [18]. Both programs can detect objects from an image by assigning objects a class and by providing a bounding box where the object is located in the image.

2.4 Knowledge graphs

An ontology is a formal representation of classes, properties, and the relations between them in a specific domain. Following the definition of Ehrlinger and Wöß, 'a knowledge graph acquires and integrates information into an ontology and applies a reasoner to derive new knowledge' [19]. Knowledge can be represented using the RDF format. This format contains tuples in the (subject, predicate, object) format. Subjects and objects correspond to concepts and the predicate describes the relationship between them. Information stored in this way is easily accessible. Several knowledge graphs are explored and seemed useful for this research [31]. However, these knowledge graphs differ in size, type, and format [31]. An example of a knowledge graph is ConceptNet which has the goal to create a commonsense knowledge base. We explored the structure of these general knowledge graphs to get an overview of how our knowledge graph should look like. However, it is decided that general knowledge graphs like ConceptNet are not integrated into our knowledge graph. This is because our research is focused on a specific environment, namely the beach environment. We will focus on knowledge graphs for autonomous robots to explore how we specifically build our knowledge graph. Looking at the review of knowledge graphs in the robotics domain, two knowledge graphs stood out of the rest [32]. These knowledge graphs are RoboBrain and KnowRob [20,21]. Both RoboBrain and KnowRob contain a clear distinction between instances and actions. They showed that this is a very useful structure of knowledge graphs for autonomous robots. These studies are however focused on generalizing a knowledge graph for autonomous robots in an indoor environment. In contrast, our research will focus on a knowledge graph for a specific outdoor environment. To infer actions from the knowledge graph, we looked into several query languages [33]. SPARQL language is eventually chosen because the language is suitable for RDF graphs and it is well documented and therefore easy in use.

3 Research Methodology

Plastic pollution is among the most commonly discussed problems the world is facing at the moment. This research is focused on removing the plastic on beaches. As discussed in the previous section, there is not one general method to remove all the plastic that is located on beaches. Furthermore, the technological possibilities of removing plastic on beaches are not well explored. We aim to set the first step towards the worldwide use of autonomous beach cleaning robots. We will look at the potential of various AI technologies, and particularly combine path planning, object detection, and knowledge graphs. These AI technologies have reached maturity and could have a positive impact on the effectiveness of the robot. We decided that we will not go into the details of the mechanical side, as others have shown that it is mechanically possible to build an autonomous plastic detection and removal beach-robot [12, 13]. Therefore, we will accept that the mechanical side of the robot is possible and we suggest the use of AI for the autonomy of the robot. Our main research question is:

How and to what extent can Artificial Intelligence support the development of an autonomous robot that detects and removes plastic on beaches?

This question is answered in four sub-parts. The sub-questions of this research are:

- ***SQ1****: How and to what extent can path planning be used for the movement of the robot on the beach?*
- ***SQ2****: How and to what extent can an object detection system be used to identify the objects that are located on the beach?*
- ***SQ3****: How and to what extent can a knowledge graph describe the environment of the beach-robot and infer actions that need to be taken?*
- ***SQ4****: Is the created system suitable for deployment?*

We contribute by setting the path towards using AI technologies to clean beaches. The idea and the extent to which the suggested system would work in real life are the main contributions of this paper.

4 Approach

Figure 1 shows the workflow of the system. It contains the tasks and corresponding outputs.

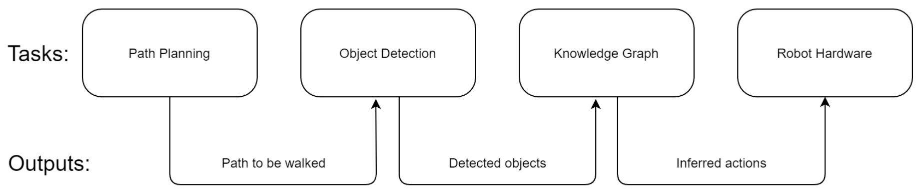

Figure 1: Workflow of the system.

Our approach follows the structure of the workflow. The idea is that path planning will provide the robot with the path that has to be walked. Therefore, we first describe how path planning can be used. Then, object detection can be used for the identification of objects. Thus, we will continue by describing which object detection program is used and how this object detection program works. The knowledge graph stores information and allows inference of actions that need to be taken based on the detected objects. So lastly, we describe how the knowledge graph is built and how queries can be used to infer the instructions for actions. The instructions for actions are then given to the robot hardware which is outside our scope of research. The code and results of this research are saved in a GitHub repository[6].

4.1 Path planning

Coverage path planning algorithms are chosen as our approach to work with path planning because covering the beach is what is needed to clean it [14]. We propose to use the boustrophedon path coverage algorithm [29]. It is focused on covering the whole beach. The goal of this research is not to program an algorithm for path planning but to show that the boustrophedon path coverage algorithm is a suitable way to provide the robot the path that has to be walked. As discussed in the related literature, the three principles of path planning should hold for this path planning algorithm [25]. First, the path that is determined has to be the shortest. Secondly, the derived path should be quickly determined. Lastly, the path should take into account that interruptions may arise. Three exceptions could occur while the robot is driving. The first is that an object has to be avoided. Examples of objects that have to be avoided are persons and animals. The second exception that could occur is that objects should be removed. Plastic bottles and plastic sacks are examples of objects that should be removed from the beach. The last exception is that the temporary bin of the robot is full and it needs to be emptied. The robot could then use object detection and the knowledge graph to infer the instructions for actions that have to be taken next to driving around. After an exception occurs, the robot continues following the simple path again. The output of this part of the system should be a path that can be walked by the robot.

4.2 Object detection

YOLO is chosen as our object detection program because YOLO can perform in real-time and with high precision [17]. We need an object detection program that can detect objects in real-time because the robot will be used in the real world. High precision of detecting objects is needed because the robot will be used among people and errors can not be afforded. YOLO sees object detection as a regression problem and uses a single neural network to predict the objects on images. YOLO uses end-to-end training on these images. YOLO works in the following way. An image is divided into several regions. Each region is weighted by the predicted bounding boxes and probabilities. From there, the objects are detected. The output of YOLO consists of a name, a bounding box, and a certainty of the predicted object. The name of the predicted object will

[6]https://github.com/Ramonprogramming/Autonomous-Beach-Cleaning-Robot

be put in a variable and this variable will be given to the knowledge graph. We will use the pre-trained model on the PASCAL VOC 2012 dataset [34]. This dataset consists of twenty object classes and the model that we use is thus trained on these items. The twenty items consist of a person class and several animals, vehicles, and indoor items. We thus chose a dataset that is not focused on a beach-specific environment. However, there were enough items that were interesting for our research so therefore we still chose to use this dataset. The goal of this research is not to find a dataset that perfectly fits our problem but to examine the performances of YOLO in a domain-specific task.

4.3 Knowledge graph

A knowledge graph is used to describe the environment and the possible actions of the robot. The distinction between objects and actions are used similarly as KnowRob and RoboBrain [20, 21]. For example, the properties of the objects are separated from the actions that need to be taken when an object is detected. Table 1 shows how the knowledge graph is built. The examples of the tuples are written down in turtle format.

Table 1: The development of the knowledge graph.

Action	**Example**
Define main objects on a beach	ex:sunbed rdf:type ex:static_object .
Define properties of main objects	ex:person ex:has_property ex:can_walk .
Define if object should be picked up	ex:bottle ex:action_required ex:pick_up .
Define how objects should be picked up	ex:bottle ex:pick_up_ability ex: one_arm .
Define properties of the robot itself	ex:robot, geo:lat '-8.659589' .

We first identify the main elements that are located at beaches. Then, we are describing the properties of these objects. We continue with defining if objects should be picked up and we proceed with elaborating how the items should be picked up. We end up with describing the properties of the robot itself. This scheme is chosen so no important aspects of the environment will be forgotten. After the knowledge graph is created, it can always be updated and revised. Figure 2 contains a visualisation of the class hierarchy of the knowledge graph. It shows that there is a clear distinction between static and dynamic objects. Static objects are objects that are not able to move or switch from a location. Plastic bottles, sunbeds, and trees are examples of static objects. Dynamic objects are objects that can move and switch from a location. Examples are humans, animals, and cars. Besides these two main classes, there is also a plastic object class that contains all the plastic objects. The plastic object class is a subclass of the static and dynamic object classes. Each class in Figure 2 is a subclass of the object class. Furthermore, every class also contains properties such as temporary information. Examples of temporary data are the location of the robot and if the temporary bin of the robot is full. These properties are not shown in the class hierarchy of the knowledge graph. The knowledge graph contains information about the beach environment as a whole and does not contain information about other environments. It is thus purely focused on the field related objects. This is because we are ultimately interested if the knowledge graph can represent the environment of the beach robot.

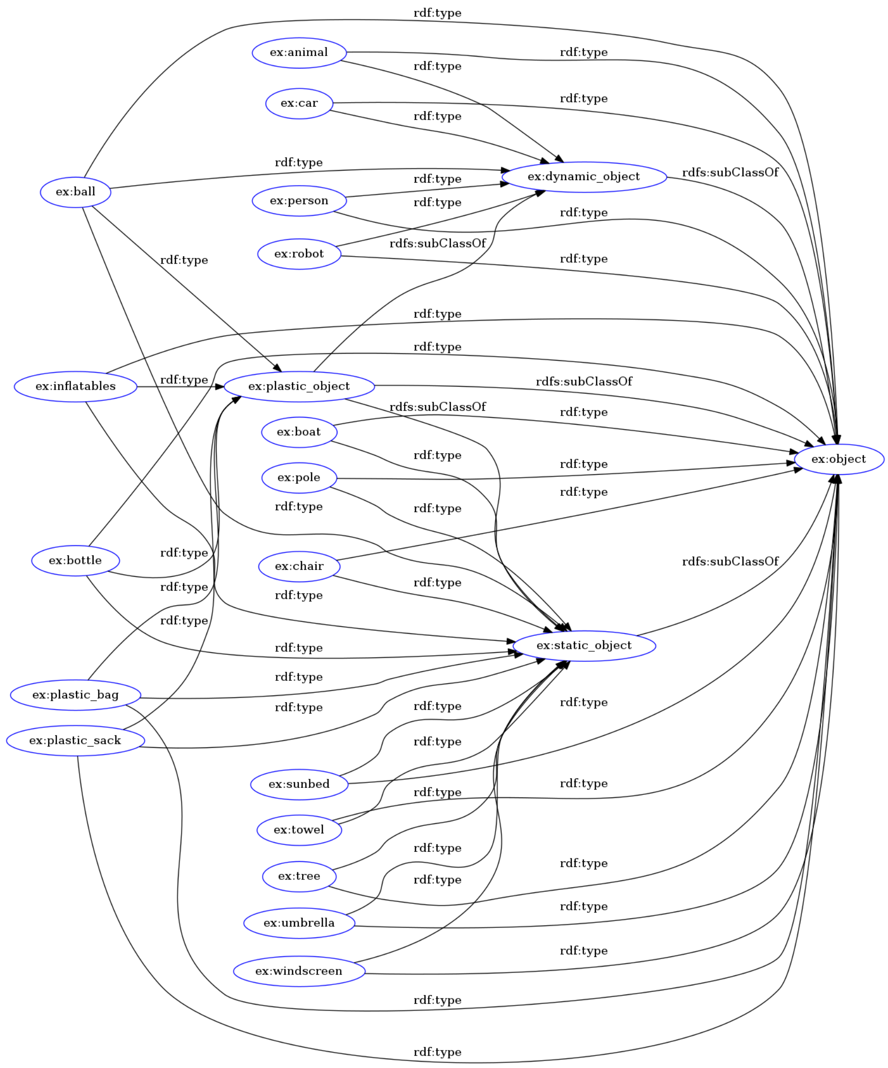

Figure 2: Class hierarchy of the knowledge graph.

Figure 3 contains a visualisation of the core of the knowledge graph. It shows how objects and properties are related to each other. We can for example see that a bottle needs to be picked up and because it has a low weight, it can be picked up with one arm. We can see the current location of the robot with the latitude and longitude information. Besides that, the temporary bin of the robot is not full at this moment. The saved location contains the last location before the robot empties the temporary bin. This location is saved because the robot then knows where it should proceed its path when the temporary bin is emptied. There are more objects, properties, and relations in the rest of the knowledge graph as described in Table 1. The visualisation of the full knowledge graph can be found in the GitHub repository[7].

[7]https://github.com/Ramonprogramming/Autonomous-Beach-Cleaning-Robot/blob/master/visualisation_knowledge_graph.png

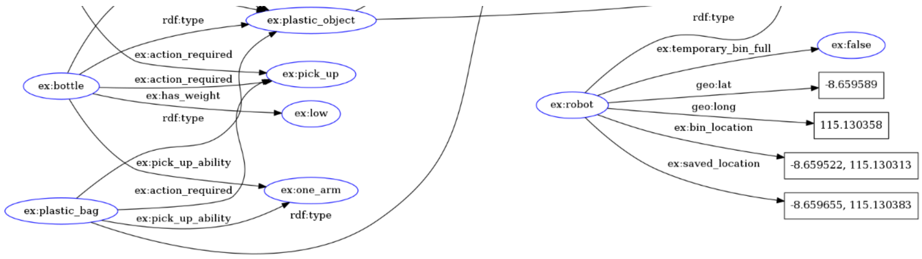

Figure 3: Visualisation of an example scenario of the knowledge graph.

Once the graph is built, it can be queried to return the instructions for actions that the robot needs to undertake. An example of such a SPARQL query can be seen in Listing 1. As we saw in the object detection section, the name of the object that is detected is saved in a variable. This variable can be inserted in the '%s' place. Then, the relevant tuples of the detected object will be queried with this SPARQL query. For example, when the system detects a person, it is implied through the knowledge graph that the person is dynamic, is able to walk and move. It is also suggested to move around the object. An additional query with the 'ex:robot' class on the '%s' place will always be run so the information of the robot will always be taken into account. This is done so that the location and the status of the temporary bin are constantly tracked. The output of both the target and the robot queries can then be given to the hardware of the robot to perform the actions that are inferred.

Listing 1: An example of a SPARQL query.

```
SELECT ?s ?p ?o            #select all tuples
WHERE {
        ?s ?p ?o
        VALUES ?s {%s} .   #where the value is the detected object
                            or the robot class
      }
```

5 Evaluation

5.1 Experimental settings

As we discussed in the approach section, we propose to use the boustrophedon path planning algorithm. We reasoned that this simple path coverage algorithm is suitable for doing the job. As discussed, we will not code the algorithm. We will evaluate if the three principles of path planning will hold for the boustrophedon algorithm to answer SQ1.

We will evaluate YOLO by looking at the performances of YOLO on 50 images of bottles and 50 images of persons. We will count how many times YOLO can correctly predict the objects on the images. Furthermore, we will try several other beach-related objects to see how YOLO performs. We will then use these results to answer SQ2 which asks to what extent YOLO can be used to identify the objects that are located on the beaches.

The capabilities of the knowledge graph are evaluated through the use of competency questions. Several competency questions that are relevant for this system are set up and checked with the knowledge graph. This is done to create awareness of what is possible and what is not. After that, several SPARQL queries are used to identify the actions that can be taken by the robot. SQ3 is answered with this evaluation method.

To evaluate the AI techniques as a whole system, a scenario is created. This scenario gives insight into the working of the system as a whole. Besides that, it shows how the subparts of the system work together. It is chosen to describe the scenario in a story so that the real-life application is visible and SQ4 can be answered.

5.2 Path planning

The visualisation of the path planning algorithm is shown in Figure 4. We will evaluate if the three principles of path planning hold. By definition, this boustrophedon path is the shortest path for covering an area [29]. The path can be quickly determined because it is a simple path. The exceptions that could occur will be handled with the object detection and knowledge graph part of the system. Thus, this boustrophedon algorithm satisfies the three principles.

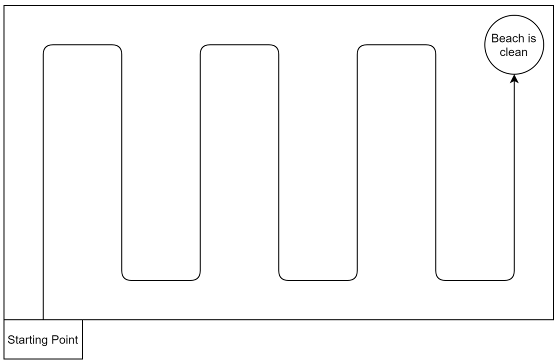

Figure 4: Simple path planning visualisation.

The movement part can also be determined with other planning algorithms. The power of object detection and knowledge graphs is that it does not matter how the robot moves. The path planning algorithm can thus be changed according to the preferences of the user. However, a simple path coverage algorithm would already satisfy the movement of the robot.

5.3 The capabilities of YOLO

The capabilities of YOLO are examined in two different ways. We did not only test several beach related images, but we also quantitatively assessed the performances on person and bottle images. As discussed in the approach, YOLO is trained on the PASCAL VOC 2012 dataset. Table 2 shows the results of the performances of YOLO on 50 images of bottles and 50 images of persons on the beach. ‘Right guess’ means that YOLO is able to identify the object, ‘no guess’ means that it does not classify the object to a class and 'wrong guess' means that YOLO assigns the wrong class to the object on the image.

Table 2: Performance of YOLO on 50 images of bottles and 50 images of persons on the beach.

	Right guess	No guess	Wrong guess	Total
Bottle	40%	50%	10%	100%
Person	100%	–	–	100%

Table 2 shows that 40% of the bottles are correctly identified as a bottle. We can also see that 10% of the bottles are assigned to the wrong class. Examples of wrong classes are the classes banana and toothbrush. Lastly, half of the bottles were not guessed at all. For the images of persons, however, it assigned all the persons to the right class. YOLO is thus capable of predicting plastic bottles but it is better at predicting persons when it is trained on this dataset. The used images mostly contained only one bottle or one person. Our results correspond to the results provided by the research on YOLO [17]. The precision of detecting objects right in their research was 22.7% for bottles and 63.5% for persons. We will now look into specific examples of detected objects to show the differences between persons and bottles. Figure 5a shows that YOLO can identify multiple persons at the same time with high precision. However, YOLO cannot detect the plastic bottles which are visible in Figure 5a. These are currently too far away and are not identified because the robot can not see them. Figure 5b shows that YOLO is also able to detect animals and even boats. YOLO does not assign general objects like the sea or the beach to a class.

(a) Several people at the beach[8]   (b) Mother, daughter and birds[9]

Figure 5: Performance of YOLO on general beach objects.

When we look at Figure 6a, we see that YOLO is able to detect plastic bottles. However, Figure 6b shows that YOLO is not able to recognize every plastic bottle that is visible on the photo. YOLO can detect plastic bottles but with less accuracy than persons. Several studies have shown that YOLO is able to work with a webcam and perform in real-time [17,35]. YOLO suits our problem very well, the only thing that has to be done is to train YOLO on a dataset specific to the beach domain. We showed that YOLO is able to perform well on images of persons but less well on images of bottles. YOLO is a suitable

[8] https://unsplash.com/photos/qH9-I3GpkQ4
[9] https://unsplash.com/photos/i–KZK__i5w

technology for the development of an autonomous plastic detection and removal beach-robot when it is trained on a beach-specific dataset.

(a) Plastic bottle on the beach[10] (b) Debris on the beach[11]

Figure 6: Performance of YOLO on plastic objects on the beach.

5.4 The potential of the knowledge graph

The abilities of the knowledge graph are evaluated with the competency questions in Table 3. The table consists of questions about the abilities of the knowledge graph, two examples per ability, and the results that we have found.

Table 3: The abilities of the knowledge graph in a beach environment.

Ability	**Examples**	**Result**
Is the knowledge graph able to represent all the static objects?	Sunbed Bottle	Yes
Is the knowledge graph able to represent all the dynamic objects?	Person Animal	Yes
Is the knowledge graph able to represent all the temporary data?	Location Temporary bin level	Yes
Is the knowledge graph able to represent all the actions that have to be taken?	Pick up Move around	Yes
Is the knowledge graph able to represent the instructions for the robot hardware?	One arm pick up Low weight	Yes
Is the knowledge graph able to contain information about multiple similar objects?	Multiple people Multiple bottles	No
Is the knowledge graph able to represent the path that needs to be walked by the robot?	Move towards a bottle Move around a human	No

Table 3 shows that the knowledge graph contains all the information about the static and dynamic objects and consists also of properties of these objects. The temporary data and the actions that need to be taken are well represented. The system is able to recognize the objects with YOLO and give the right actions

[10]https://unsplash.com/photos/4xmgrNUbyNA
[11]https://unsplash.com/photos/FMrZLPdDyx4

that the system needs to take with the knowledge graph. The actions that need to be taken from the knowledge graph are received with SPARQL queries.

The limitations of the knowledge graph are the following. The knowledge graph treats similar objects in the same way. For example, when YOLO detects a bottle, the knowledge graph only contains information about the bottle. It can not infer if the bottle is full or empty. However the advantage is that you can easily add new information. Another limitation of the knowledge graph is that it is not able to represent the path that the robot needs to walk towards or around objects. Besides some limitations, a knowledge graph is a suitable method for the representation of the environment of the robot and the instructions for actions can be inferred with SPARQL queries.

5.5 Real-life scenario

In this section, we will describe a scenario of how the full system works in real life. Figure 7 contains a visualisation of the simple path planning algorithm, exceptions that could occur, and the most important tuple that is inferred through the knowledge graph.

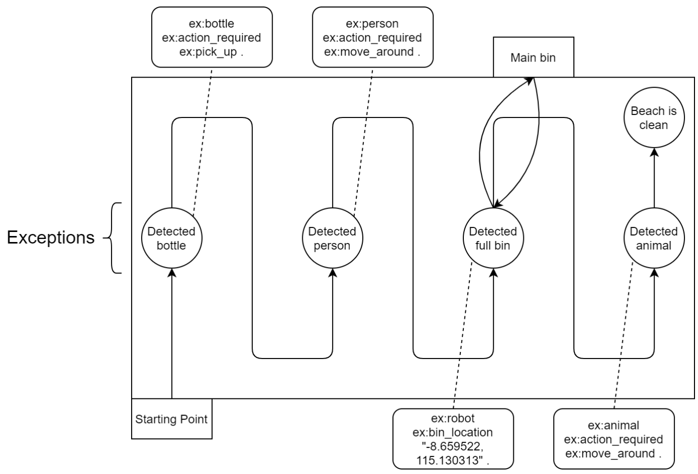

Figure 7: A scenario of the working of the system on the beach.

The robot starts by following the simple path. At a certain point, a bottle is detected with YOLO. The SPARQL query in Listing 2 is run to receive the information about the bottle through the knowledge graph. The knowledge graph gives back that the plastic bottle needs to be picked up and because the plastic bottle has a low weight, it can be picked up with one arm.

Listing 2: SPARQL query to receive information about the detected bottle.

```
SELECT ?o
WHERE {
        {ex:bottle ex:action_required ?o .}     #output: pick_up
        UNION
        {ex:bottle ex:has_weight ?o .}          #output: low
        UNION
        {ex:bottle ex:pick_up_ability ?o .}     #output: one_arm
      }
```

This information is given to the hardware of the robot. When the action is performed, the robot continues on the path. From there, it proceeds its way until, at a certain point, a person is detected with YOLO. It is inferred with the SPARQL query in Listing 3 that the person is able to move. The action required is to move around the person and proceed with following the path.

Listing 3: SPARQL query to receive information about the detected person.

```
SELECT ?o
WHERE {
        {ex:person ex:action_required ?o .}    #output: move_around
        UNION
        {ex:person ex:has_property ?o .}       #output: can_move
      }
```

At another point in time, the temporary bin of the autonomous robot is full and needs to be emptied. This triggers the SPARQL query in Listing 4 that gives the location of the main bin. The current location is saved in the knowledge graph and the robot proceeds its way towards the main bin location. Actions about plastic objects are switched off during the emptying of the temporary bin. This is done so that no plastic will be picked up because the temporary bin is already full. The robot empties the bin and moves back to the saved location.

Listing 4: SPARQL query to receive information about the robot.

```
SELECT ?o
WHERE {
        {ex:robot ex:temporary_bin_full ?o .}   #output: true
        UNION
        {ex:robot ex:bin_location ?o .}         #output:
                                                 -8.659522,
                                                 115.130313
      }
```

After this action, an animal is detected with YOLO. The SPARQL query in Listing 5 outputs that this animal is a dynamic object and needs to be avoided.

Listing 5: SPARQL query to receive information about the detected animal.

```
SELECT ?o
WHERE {
        {ex:animal ex:action_required ?o .}    #output: move_around
        UNION
        {ex:animal ex:has_property ?o .}       #output: dynamic
      }
```

After all these exceptions, the whole path has been walked and the beach is clean. This is how the system works in a real-life scenario.

6 Discussion

The path planning part of the research showed that only a simple path planning algorithm would do the job. The boustrophedon path coverage algorithm is the shortest path to cover an area, can be quickly determined, and is adaptable to exceptions. Therefore, it satisfies the three principles of path planning. Future research could focus on the development and testing of this algorithm for a beach environment. It can then be seen how it works in real life.

Our results showed that YOLO is able to detect objects with high precision. This corresponds to the presentation of YOLO itself [17]. However, we used the off-the-shelf PASCAL VOC 2012 dataset, and therefore our dataset lacked beach-specific information [34]. The developers of YOLO have shown that YOLO is better at detecting persons than bottles [17]. Our results showed that YOLO is able to detect every person, even when the person is in low resolution or in the background of the image. In contrast, it was harder for YOLO to detect plastic bottles. We can conclude that training could provide YOLO with better detection of bottles as well. The number of objects that could be detected by YOLO on the beach could be improved by training a model on multiple images and research has shown that this will contribute largely to the performances of the system [35]. Thus, the strength of YOLO is proven and future research could focus on creating a more suitable dataset. Future research could also focus on real-life exceptions. For example, if YOLO detects a person holding a bottle, it is questionable if the bottle needs to be picked up.

The connection of YOLO to the knowledge graph seems to be a good choice as well. The objects can be detected with a webcam and YOLO gives the names of the detected objects back to the system. SPARQL queries can then be used on the knowledge graph to infer the instructions for actions that have to be taken. The graph also showed its potential but it has not yet reached its full potential. We saw that a knowledge graph can be used to identify everything that is located at a beach. Furthermore, the knowledge graph is currently able to infer if items have to be picked up and how this should be done. Future research could find out how the knowledge graph should represent the movement towards objects. Furthermore, future research could look into temporary data about the process. Examples that could be added to the temporary data are time and place data. Then, the robot would know at what time and place it found an object.

The scenario summarized that these AI techniques support the development of an autonomous robot that detects and removes plastic on beaches. For the system as a whole, future research could focus on linking the AI technologies proposed in this research with the mechanical side and testing of the system on a real beach.

7 Conclusion

This paper attempts to answer the question to what extent Artificial Intelligence supports the development of an autonomous robot that detects and removes plastic on beaches. Plastic pollution is a problem that affects the whole world and an autonomous plastic detection and removal beach-robot could be used to remove plastic on beaches more efficiently. This autonomous robot was built with three AI techniques: a path planning part, an object detection part called YOLO, and a knowledge graph. The path planning part of the autonomous robot can be a simple path coverage algorithm. We concluded that this part can be adjusted to preferences but we recommend keeping it simple. For the object detection part, we saw that YOLO is able to detect objects with high precision. We showed that YOLO is better at detecting persons than bottles because we used a dataset that is not specifically focused on a beach environment. If YOLO is trained on a dataset that consists of all the elements that are known around the beach, it would reach the full potential of object detection. We can conclude that YOLO is suitable for the development of the autonomous beach-robot. The knowledge graph can represent all the static and dynamic objects that are located on beaches. It can describe if objects need to be avoided or need to be picked up. Furthermore, it describes that plastic objects need to be picked up in a specific manner. The knowledge graph is also able to represent locations. The outcome of SPARQL queries that are run on the knowledge graph provides the robot with instructions for the actions that need to be taken. This can then be given to the robot hardware. The real-life scenario showed that the three AI techniques can work together to perform the desired actions. We can conclude that AI can support the development of an autonomous robot that detects and removes plastic on beaches to a large extent.

References

[1] D. K. Barnes, F. Galgani, R. C. Thompson, and M. Barlaz, "Accumulation and fragmentation of plastic debris in global environments," *Philosophical Transactions of the Royal Society B: Biological Sciences*, vol. 364, no. 1526, pp. 1985–1998, 2009.

[2] J. G. Derraik, "The pollution of the marine environment by plastic debris: a review," *Marine pollution bulletin*, vol. 44, no. 9, pp. 842–852, 2002.

[3] T.-K. Liu, M.-W. Wang, and P. Chen, "Influence of waste management policy on the characteristics of beach litter in kaohsiung, taiwan," *Marine Pollution Bulletin*, vol. 72, no. 1, pp. 99–106, 2013.

[4] H. Jayasiri, C. Purushothaman, and A. Vennila, "Quantitative analysis of plastic debris on recreational beaches in mumbai, india," *Marine pollution bulletin*, vol. 77, no. 1-2, pp. 107–112, 2013.

[5] C. Moore, G. Lattin, and A. Zellers, "Quantity and type of plastic debris flowing from two urban rivers to coastal waters and beaches of southern california," *Revista de Gestão Costeira Integrada-Journal of Integrated Coastal Zone Management*, vol. 11, no. 1, pp. 65–73, 2011.

[6] T. R. Merrell Jr, "Accumulation of plastic litter on beaches of amchitka island, alaska," *Marine Environmental Research*, vol. 3, no. 3, pp. 171–184, 1980.

[7] K. J. McDermid and T. L. McMullen, "Quantitative analysis of small-plastic debris on beaches in the hawaiian archipelago," *Marine pollution bulletin*, vol. 48, no. 7-8, pp. 790–794, 2004.

[8] M. Schneier, M. Schneier, and R. Bostelman, *Literature review of mobile robots for manufacturing.* US Department of Commerce, National Institute of Standards and Technology, 2015.

[9] B. Preising, T. Hsia, and B. Mittelstadt, "A literature review: robots in medicine," *IEEE Engineering in Medicine and Biology Magazine*, vol. 10, no. 2, pp. 13–22, 1991.

[10] N. V. Reddy, A. V. V. Reddy, S. Pranavadithya, and J. J. Kumar, "A critical review on agricultural robots," *International Journal of Mechanical Engineering and Technology*, vol. 7, no. 4, pp. 183–188, 2016.

[11] P. Sapaty, "Military robotics: latest trends and spatial grasp solutions," *International Journal of Advanced Research in Artificial Intelligence*, vol. 4, no. 4, pp. 9–18, 2015.

[12] T. Ichimura and S.-i. Nakajima, "Development of an autonomous beach cleaning robot "hirottaro"," in *2016 IEEE International Conference on Mechatronics and Automation*. IEEE, 2016, pp. 868–872.

[13] S. Watanasophon and S. Ouitrakul, "Garbage collection robot on the beach using wireless communications," *Int. Proc. Chem. Biol. Environ. Eng*, vol. 66, pp. 92–96, 2014.

[14] E. Galceran and M. Carreras, "A survey on coverage path planning for robotics," *Robotics and Autonomous systems*, vol. 61, no. 12, pp. 1258–1276, 2013.

[15] H. S. Parekh, D. G. Thakore, and U. K. Jaliya, "A survey on object detection and tracking methods," *International Journal of Innovative Research in Computer and Communication Engineering*, vol. 2, no. 2, pp. 2970–2978, 2014.

[16] L. Liu, W. Ouyang, X. Wang, P. Fieguth, J. Chen, X. Liu, and M. Pietikäinen, "Deep learning for generic object detection: A survey," *International journal of computer vision*, vol. 128, no. 2, pp. 261–318, 2020.

[17] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "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.

[18] R. Girshick, "Fast r-cnn," in *Proceedings of the IEEE international conference on computer vision*, 2015, pp. 1440–1448.

[19] L. Ehrlinger and W. Wöß, "Towards a definition of knowledge graphs." *SEMANTiCS (Posters, Demos, SuCCESS)*, vol. 48, 2016.

[20] A. Saxena, A. Jain, O. Sener, A. Jami, D. K. Misra, and H. S. Koppula, "Robobrain: Large-scale knowledge engine for robots," *arXiv preprint arXiv:1412.0691*, 2014.

[21] M. Tenorth and M. Beetz, "Knowrob—knowledge processing for autonomous personal robots," in *2009 IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE, 2009, pp. 4261–4266.

[22] T. Kataoka, H. Hinata, and S. Kako, "A new technique for detecting colored macro plastic debris on beaches using webcam images and cieluv," *Marine pollution bulletin*, vol. 64, no. 9, pp. 1829–1836, 2012.

[23] Z. Ge, H. Shi, X. Mei, Z. Dai, and D. Li, "Semi-automatic recognition of marine debris on beaches," *Scientific reports*, vol. 6, no. 1, pp. 1–9, 2016.

[24] C. Battisti, G. Poeta, L. Pietrelli, and A. T. Acosta, "An unexpected consequence of plastic litter clean-up on beaches: too much sand might be removed," *Environmental Practice*, vol. 18, no. 4, pp. 242–246, 2016.

[25] N. Sariff and N. Buniyamin, "An overview of autonomous mobile robot path planning algorithms," in *2006 4th Student Conference on Research and Development*. IEEE, 2006, pp. 183–188.

[26] X. Sun and Y. Zhang, "A review of domain knowledge representation for robot task planning," in *Proceedings of the 2019 4th International Conference on Mathematics and Artificial Intelligence*, 2019, pp. 176–183.

[27] F. Gul, W. Rahiman, and S. S. Nazli Alhady, "A comprehensive study for robot navigation techniques," *Cogent Engineering*, vol. 6, no. 1, p. 1632046, 2019.

[28] R. Gayathri and V. Uma, "Ontology based knowledge representation technique, domain modeling languages and planners for robotic path planning: A survey," *ICT Express*, vol. 4, no. 2, pp. 69–74, 2018.

[29] H. Choset and P. Pignon, "Coverage path planning: The boustrophedon cellular decomposition," in *Field and service robotics*. Springer, 1998, pp. 203–209.

[30] D. K. Prasad, "Survey of the problem of object detection in real images," *International Journal of Image Processing (IJIP)*, vol. 6, no. 6, p. 441, 2012.

[31] J. Yan, C. Wang, W. Cheng, M. Gao, and A. Zhou, "A retrospective of knowledge graphs," *Frontiers of Computer Science*, vol. 12, no. 1, pp. 55–74, 2018.

[32] S. Wang, Y. Zhang, and Z. Liao, "Review on the knowledge graph in robotics domain," in *3rd International Conference on Computer Engineering, Information Science & Application Technology (ICCIA 2019)*. Atlantis Press, 2019.

[33] P. T. Wood, "Query languages for graph databases," *ACM Sigmod Record*, vol. 41, no. 1, pp. 50–60, 2012.

[34] M. "Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman, ""the {PASCAL} {V}isual {O}bject {C}lasses {C}hallenge 2012 {(VOC2012)} {R}esults"," "http://www.pascal-network.org/challenges/VOC/voc2012/workshop/index.html".

[35] M. Fulton, J. Hong, M. J. Islam, and J. Sattar, "Robotic detection of marine litter using deep visual detection models," in *2019 International Conference on Robotics and Automation (ICRA)*. IEEE, 2019, pp. 5752–5758.