Product: BinButler
Team: BinBuilders

Abstract

The BinButler is an assistive robot featuring a bin mounted on a wheelbase. The bin is divided into two compartments each featuring automatic lids for recyclable and non-recyclable waste, as well as sensors to detect the fullness of each compartment. Upon a user's request, the BinButler autonomously navigates to the requesting user allowing them to dispose of different types of waste.

The BinButler is accompanied by a web app which sends precise goals for movement and operation. It also features a fully customised image classification model to advise users on how to dispose of their waste as well as information about the bin's fullness. Overall, these features of BinButler intend to simplify waste handling in professional environments.

1. Project management update

1.1. Unaccomplished Goals

1. Implementing a lift system that allows for changing out bin types off custom racks (Failed)

2. Implementing bin fullness detection (Partially Achieved)

1.2. Deviations

The initial plan for BinButler was to have a smart movement base that could lift and pick up certain bins and deliver them to users. By Demo 2, a prototype of the scissor lift mechanism had been manufactured. However, upon testing it with a high torque (1.26 Nm) stepper motor (RS, 2024), the mechanism struggled to lift an empty bin, hence a significant redesign of the project was required. Upon reflection, it became apparent that the BinButler should allow users to discard different types of waste in one trip by containing multiple bins on one wheelbase - one for recycling and another for general waste. Development therefore shifted from the lifting mechanism to creating a robot with multiple bins. Development was also able to be redirected to both incorporating a waste classification model and animations synchronised across the web app and robot for user feedback.

The fullness detection system has been classified as partially achieved due to the current system only detecting if the bin is 33%, 66%, or 100% full. Ultrasonic sensors were initially used for fullness detection. The output from these sensors was too noisy to provide an accurate measurement due to echoes within the bin. Hence, these sensors were replaced with infrared sensors for the final demo. While noise was reduced with these sensors, they were still too noisy to provide accurate readings, hence their readings were further discretised.

Leading up to the final demo the web application was redesigned for ease of use. Descriptive text was added to major functions to guide the user through processes. The interface was redone to fit onto all sizes of screens without scrolling. These improvements have led to a Google Lighthouse score of 94 as seen in Figure 5 in Appendix A.1.4.

1.3. Organisation

At the start of the project, three teams were created: movement, web, and peripherals. The movement team was in charge of writing the code surrounding the wheelbase and its movement. This included setting up a ROS environment and communicating between the wheelbase and a host server. The web team was in charge of the user interface and a web server which talks to other services in our system. The peripherals team was in charge of the physical components of the BinButler, including the lid and fullness sensors. The peripherals team also worked on the lift mechanism and bin 'grabbers' that haven't been used in the final system.

A GitHub repository was used to keep track of all code in the project. Each team had a sub-folder administrated by someone in the team. The code was written and reviewed on separate branches before being merged into the main branch to avoid any issues. A shared Google Drive was used to share non-code documents and WhatsApp was used for team-wide and inter-team communications.

1.4. Team Members

Guy (Web/Presenter) - Designed and implemented the web application and assisted in integrating all other services with the web server. Additionally set up Docker for a micro-service architecture. (250 hours)

Ismaeel (Movement/Image Model/Animations/Presenter) - Set up the ROS environment, autonomous mapping of an environment, and autonomous navigation. Created a custom Docker Network. Designed and trained the image classification models. Developed the animation system. Integrated all these systems into the web server. Assisted in the redesign of the web app. Wrote up the QA section of the report and a detailed report on the image classification process. Developed code to integrate the lid mechanism into the web server. Assisted in the writing of reports, the creation of slides and industry posters, and presentations. (350 hours)

Nilesh (Movement/Peripherals/Presenter) - Assisted with setting up ROS and basic movement on the TurtleBot. Implemented and fine-tuned fullness detection mechanism and helped integrate this with the web app. Composed the presentation and slides for each demo, including performing market research. Liaised with the client for demo feedback. Guided in producing the industry poster and redesigning the system after demo two. (200 hours)

Francisco (Peripherals) - Assisted with sensor implementation and performed the QA for this. Provided general assistance around teams. Assisted in the writing of reports and the creation of slides. (120 Hours)

Luke (Movement) - Worked on initial Raspberry Pi/ROS setup, provided general assistance to teams and wrote and edited the demo reports. (120 Hours)

Stan (Peripherals) - Designed and engineered the lid-opening mechanism and integrated it into the system. Worked on the redesign of the dual bin system. (200 Hours)

John (Peripherals/Presenter) - Designed the bin's structural components, produced animations, and assembled components. Worked on the redesign of the dual bin system. Assisted in the design of the industry poster and demo slides. Created custom wiring for the hardware. Designed and prototyped the lift mechanism before it was redesigned. Assisted in the redesign of the web app. (250 Hours)

Efe (Peripherals/Web) - Worked on the initial design and software implementation of the lid system. Assisted in the user-focused design and industry day poster design, as well as editing and assisting in creating the videos for the presentation. (120 hours)

1.5. Budget

Table 1 shows how the monetary budget of £300 was used. Due to the loss of some files, estimates were provided for the amount of material used. The current design no longer utilises the purchased motor.

Table 2 shows how the allocated 10 hours of technician time was used. There were multiple cases in which parts were re-manufactured due to unforeseen issues with tolerance and general mistakes in modelling. More careful design and engineering could have reduced the used technician time.

Item	Cost/Unit (£)	Quantity	Cost (£)
Stepper motor	46.37	1	46.37
6mm MDF	8/sheet	~1.680	13.44
9mm MDF	11/sheet	~0.100	1.10
3mm Plywood	18.25/sheet	~0.605	11.04
PLA filament	0.025/g	~300	7.50
PVA filament	0.115/g	~100	11.50
Misc	-	-	10
		Total	£100.95
		Remaining	**£199.05**

Table 1. Monetary budget usage

	#3D prints (15 min)	#Laser cuts (15 min)	Time of laser cuts (min)	Misc (min)	Time usage (min)
Demo 1	0	4	8	60	128
Demo 2	9	4	8	0	203
Demo 3	6	8	20	15	245
				Total	576 min
				Remaining	**24 min**

Table 2. Technician time budget usage

1.6. Post-mortem

Stengths: The robot's movement was achieved early in the project. This prevented critical issues found by other teams regarding ROS. The waste classification system was successfully tailored for BinButler's needs without loss of accuracy providing an edge over competitors. Dockerising the entire system is novel compared to other teams and provides great stability.

Weaknesses: A lack of concrete deadlines set internally by the team caused challenges. Work was rushed near the demo deadlines due to this. After success with the robot's path planning for Demo 2, this aspect of the system was neglected and no further optimisations were made. Additionally, the bin's design did not consider wiring for electronic components, leading to a messy-looking final product. Finally, a significant portion of time was lost in redesigning old systems as other components could not progress until the structure was redesigned. The purchase of a £46.37 stepper motor was not utilised due to the redesign.

2. Quantitative analysis and testing

QA was split into 5 major parts: navigation, software, hardware, image classification, and end-to-end testing.

2.1. Navigation Quality Assurance

The quality of path planning was evaluated by travel time, accuracy, and precision. The goal of this system is to navigate within a 5cm error radius of a user's position at any point on the map in less than a minute.

2.1.1. Travel Time

Goal Distance (m)	Time Taken (seconds)
1.0	9.76
1.0	12.28
1.5	11.70
1.5	18.89
2.0	20.37
2.0	23.53
2.5	35.10
2.5	31.05
3.0	33.53
3.0	39.26

Table 3. Time Evaluation

The time taken to travel was calculated by setting the BinButler at its start position in the testing area and timing how long it takes to arrive at locations of different distances away.

Justification: This directly measures the path planner's efficiency when navigating to users. A lower response time provides better user satisfaction when using the BinButler and more efficient use of the onboard battery.

Evaluation: Looking at Table 3 it can be seen that the planning system efficiently navigated to users under the specified goal of 1 minute. With the average travel time being 23.55 seconds, this system can be relied on to navigate quickly and increase user satisfaction.

2.1.2. Accuracy

Goal Distance (m)	Error Radius (cm)
1.0	2.46
1.0	2.10
2.0	2.67
2.0	2.34
3.0	2.61
3.0	2.51

Table 4. Accuracy Evaluation

The accuracy of the BinButler's path planning was measured by putting a physical navigation goal in the testing area and measuring the distance between the closest point of the BinButler and the centre of the goal.

Justification: This directly measures the path planner's accuracy when navigating to users. The closer the BinBulter can navigate to a user, the quicker the user can put their waste in the bin.

Evaluation: Table 4 shows the path planning system accurately navigating to users within an average of 2.45cm. This clears the set goal.

2.1.3. Precision

Goal Distance (m)	Test 1 Error (cm)	Test 2 Error (cm)	Test 3 Error (cm)	Std Dev (cm)
1.0	2.45	2.37	2.29	0.08
2.0	2.27	2.27	2.30	0.01
3.0	2.53	2.41	2.57	0.08

Table 5. Precision Evaluation

The precision of the BinButler's path planning was measured by putting a physical navigation goal in the testing area and measuring the distance between the BinButler's position when navigating to these goals three times.

Justification: This directly measures the path planner's precision when navigating to users. The more similar the BinButler travels, the more reliance users can have on the system and comfortably use it.

Evaluation: Table 5 displays the path planning system consistently navigating to users within an average of 0.06cm.

2.2. Software Quality Assurance

To ensure the reliability and efficiency of the web server and application, a software testing approach encompassing unit tests, integration tests, and a continuous integration (CI) pipeline was used (showcased in Figure 4 in appendix A.1.3). The goal for this system is to pass all functional test cases.

2.2.1. Unit Testing

Justification: Unit tests were designed to verify the correctness of individual functions and components within both the web server and application. By testing these components in isolation, quickly identifying and rectifying any logical or runtime errors was simplified. This granular level of testing was crucial for maintaining code quality and ensuring each component performed as expected.

Evaluation: Unit tests covered critical functionalities such as user authentication, data processing, and API request handling. Table 11 in Appendix A.1.1 demonstrate all functions and methods passing 100% of the test cases. A high degree of code coverage was also found at 96%. These results indicate a robust foundation for the web server and application, minimising the risk of functional errors in production.

2.2.2. Integration Testing

Justification: Integration tests were conducted to evaluate the interactions between different components and services within the system. This testing phase was essential for verifying the correct data flow and cooperation between micro-services, including database interactions, service-to-service communication, and end-to-end data processing.

Evaluation: The integration testing phase highlighted several critical areas where communications were failing. Adjustments were made to how images are encoded and sent to the backend image classification service. This resulted in 100% of tests passing as shown in Table 12 in Appendix A.1.2. These tests ensured the micro-service architecture functions cohesively and provide a seamless experience for users.

2.2.3. User focused design and testing

Justification: The web application must be as accessible as possible for new users to allow a wide range of users to use the application. Following user-focused design principles allows for easy and effective use of the application. A descriptive UI/UX approach removed the need for separate user guides.

For the web application, user focused descriptions inspired a call-to-action principle in which the user can use the application without decision paralysis (Sproutsocial, 2023).

Evaluation: Feedback received regarding summoning the bin to a selected location from the web app led to including small and effective descriptions of web app features. Another beneficial piece of feedback led to re-scaling certain features on the web application interface to fit everything on one screen. Feedback found from Google Lighthouse led to improvements in web application performance and accessibility leading to scores of 100% and 94% respectively as shown in Figure 5 found in appendix A.1.4.

2.3. Image Classification Quality Assurance

Testing metrics and methods of the waste classification model have been tailored to assess goals of accuracy, preciseness, and effectiveness. These goals were evaluated using a combination of confusion matrices and key performance indicators which include accuracy, precision, recall, and an F1 score.

Accuracy refers to the total number of correct identifications while precision encompasses how many items classified within a class were correctly identified. The recall relates to how many items of a certain class were identified correctly. While F1 is a combination of accuracy and recall, F1 is the major metric used to assess how well the model avoids false positives and false negatives.

Confusion matrices provided visualisations of classifications of items and how they might be misidentified and allowed for fine-tuning of the Vision-Transformer (ViT) model.

2.3.1. Results

Category	Accuracy (%)	Precision	Recall	F1
Total	90.57	0.905	0.905	0.905
Cardboard	91.73	0.902	0.917	0.909
Ewaste	94.70	0.934	0.947	0.940
Glass	91.34	0.931	0.913	0.922
Medical	91.37	0.918	0.913	0.916
Metal	90.38	0.910	0.903	0.907
Paper	87.03	0.873	0.870	0.872
Plastic	87.31	0.870	0.873	0.872

Table 6. Accuracy, Precision, Recall and F1 score for the material classification model testing.

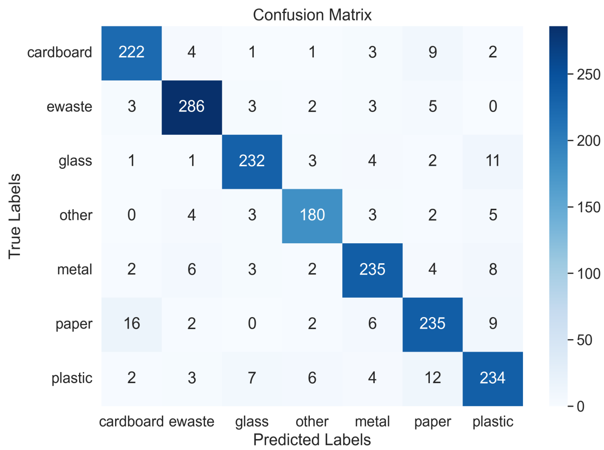

Figure 1. Confusion Matrix results when testing the material classifier

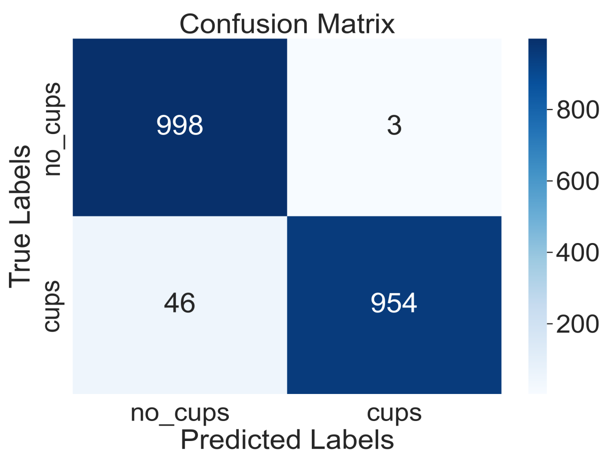

Figure 2. Confusion Matrix results when testing the cup classifier

Category	Accuracy (%)	Precision	Recall	F1
Total	97.55	0.976	0.975	0.975
Cup	95.40	0.996	0.954	0.974
No Cup	99.70	0.955	0.997	0.976

Table 7. Accuracy, Precision, Recall and F1 score for the cup classification model testing.

Figure 1 shows a confusion matrix of the waste material model when evaluated on test data. Clear signs of success can be seen barring a few outliers or misclassification of similar materials. Table 6 displays metrics of the testing process. Referencing the table it can be seen that accuracy and F1 are very similar - displaying a small percentage of false positives and negatives. Figure 2 and Table 7 display the same metrics but for the cup classification model. Similar values including an accuracy of 95%+ can be found and similar conclusions can be drawn.

2.3.2. Evaluation

Based upon the results above the model was deemed suitable for classifying waste management in the web application. The results found in Tables 6 and 7 provide confidence that users can trust and learn from the AI model.

2.4. Hardware Quality Assurance - Lid Mechanism

A series of tests were conducted focusing on durability and response time to ensure the functionality and reliability of the bin's lid mechanisms. These tests were critical for assessing the practicality of the bin in real-world scenarios where both speed and precision are key to user satisfaction. The goal of this system is to have both lids open and close 10 times in a row with a response time of 2 seconds for each request.

2.4.1. Durability

Justification: It is imperative for the BinButler's mechanisms to withstand daily wear-and-tear as to keep repair costs down.

Evaluation: Durability tests involved cyclic opening and closing of both lids for a predefined number of cycles aiming to simulate long-term usage within a short period. The purpose was to identify any potential wear or failure points in the mechanisms. After simulating 100 requests worth of opening and closing cycles, both lids showed minimal wear and continued to operate within acceptable parameters. This demonstrated the system's durability and long-term reliability.

2.4.2. Response Time

Justification: Testing response time gives an idea of the average time-waiting a user would have to endure and is part of the bin's overall efficiency.

Trial	Recycling Lid Time (s)	General Waste Lid Time (s)
1	1.4	1.7
2	1.5	1.6
3	1.6	1.5
4	1.5	1.7
5	1.5	1.6
Average	**1.5**	**1.6**

Table 8. Response times for recycling and general waste lids

Evaluation: The response time for each lid's opening and closing was measured from the moment a command was issued until the lid had closed. From Table 8, the average response time for the recycling lid was 1.5 seconds while the general waste section lid was 1.6 seconds. These results indicate a swift reaction to user commands ensuring a seamless user experience.

2.5. Evaluation

Looking at these results the lid mechanisms can be seen to service our users with a quick enough response time and durability to increase user satisfaction.

2.6. Hardware Quality Assurance - Fullness Sensors

2.6.1. Distance Testing

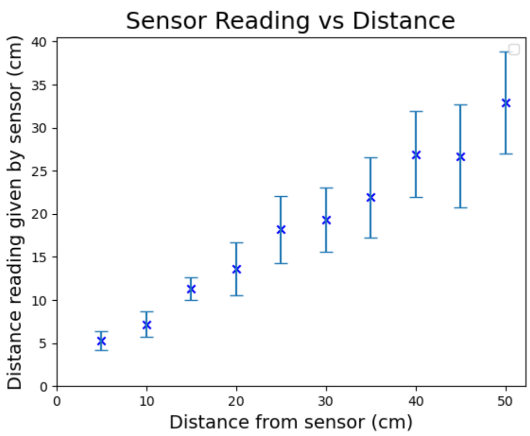

Figure 3. Sensor Distance Test Results

Distance tests were performed to test the fullness sensors. These tests involved measuring real-world distance and sensor outputs at various fullness levels. This provides information on the accuracy and precision of the sensors.

Justification: The fullness sensor system integrated into the lid has been designed to notify appropriate personnel when a bin needs to be emptied. Incorrect readings could lead to bins requesting emptying too often or too few times.

Evaluation: Figure 3 reveals significant variances between actual distances and sensor readings, hindering precise measurements. Nonetheless, a rough estimation of bin fullness was achievable, with sensor outputs declining non-linearly as the bin filled. Key fullness indicators were set at 0%, 33%, 66%, and 100% levels. Precise bin capacity isn't critical for operations; crucially, it's the near-full (approximately 100%) indication that necessitates action. Hence, the sensors' performance was considered sufficient for our needs, as detailed in Table 13 in appendix A.2.1.

2.7. End-to-End Quality Assurance

Justification: The purpose of end-to-end testing is to simulate real-world use cases to ensure that the system operates as expected in practical scenarios. This helps identify issues such as system-wide data flow problems, user interface issues, and overall performance under normal usage conditions.

Scenarios: The full end-to-end tests included: (1) A single user requesting a bin, (2) multiple users requesting a bin, (3) a user requesting a bin and the path back being blocked.

Results:

Scenario	Expected Result	Achieved?
1	Successful request	✓
2	Successful requests	✓
3	Notify cleaners of failure	✓

Table 9. End-to-End Testing Outcomes

Evaluation: The end-to-end testing confirmed that the BinBulter system performed effectively in a real-world scenario. This phase of testing was instrumental in providing confidence in the system's ability to deliver a seamless and reliable waste disposal service.

3. Estimated system budget

Using an estimated average salary of software, robotics engineers, and product designers of £42,953 (Indeed, 2024a)(Indeed, 2024b)(Indeed, 2024c) and interpolating to £20.65/hr, the estimated labour cost of the project is ~£31,844. This number includes team member's contributions as well as technician time (priced at £25/hour).

Table 10 shows that one BinButler unit costs £1457.61. In total, accounting for labour costs, monetary budget usage, and the cost of the robotic unit, our system's budget would fall ~ £34,000.

Item	Cost / Unit(£)	Quantity	Cost(£)
TurtleBot Waffle	-	1	1275.8
DC motors	2.8	2	5.60
IR sensors	9.50	2	19
7in Screen	-	1	68
Rasp. Pi 3B+	-	1	35
Grove Shield	-	1	7
Powerbank	-	1	10
Motor board	- 1	10	
3mm Plywood	18.25/sheet	0.046	0.84
6mm MDF	8/sheet	0.351	2.81
PLA filament	0.025/g	50	1.25
Misc	-	-	10
		Total	**£1457.61**

Table 10. Robot Cost

4. Miscellaneous

4.1. Waste Classification Report

4.1.1. The Model

The BinButler model was split into two steps, similar to a recent waste management paper (Majchrowska et al., 2022) but with one major alteration. The first step of object detection was replaced with a unique object classification. This step classifies any items which might not be fit to be recycled even if made of a recyclable material. For example, coffee cups may sometimes require manual checking of the inside of the cup in case there is a plastic lining. This change was made to ensure proper recycling practices and to inform users about certain quirks of recycling.

A diagram of our model architecture can be found in Appendix A.3.5

The waste management paper outlines a two-step waste categorisation process: detection and classification. Detection involves creating bounding boxes around objects, which are then cropped for classification. The paper updates the classification method by adopting a Vision Transformer (ViT) model, enhancing global attention capture without the limitations of kernel sizes, as opposed to traditional convolutional models (Majchrowska et al., 2022; Dosovitskiy et al., 2020). To mitigate the ViT model's computational demands, the approach involves dividing images into 16x16 patches, significantly reducing processing requirements by focusing attention on these patches.

4.1.2. Data

For waste material classification, a ViT model was fine-tuned with the Trashbox dataset, classifying waste like cardboard, e-waste, and plastic, relevant to office spaces. The dataset, designed for another study, includes an 80-10-10 split for training, testing, and validation, with preliminary checks for image integrity (Kumsetty et al., 2022). Additionally, object detection for identifying coffee cups was trained using a custom dataset alongside a recyclable objects dataset, distinguishing between "Cup" and "No Cup" across 30k+ images with similar data splits. This dataset amalgamates images pertinent to both generic waste and specific items like coffee cups, facilitating comprehensive waste management (Marionette, 2022; Sekar, 2019).

Training Accuracy and Loss can be seen in appendix A.3.

A. Appendix

A.1. Software Testing

A.1.1. Unit Test Results

Test Type	Pass Rate
Web App Tests	
User Authentication Tests	100%
Database Update Tests	100%
Database Delete Tests	100%
Database Insert Tests	100%
User Login Tests	100%
User Logout Tests	100%
User Registration Tests	100%
Web Server Tests	
Data Message Parsing Test	100%
Image Data Parsing Test	100%
Map Parsing Data Test	100%
Communication to Server Tests	100%
Map Request Test	100%
Image Classification Request Test	100%

Table 11. Unit Tests Summary

A.1.2. Integration Test Results

Test Type	Pass Rate
Web App Integration Tests	
Session Management	100%
Session Cookie Creation	100%
Session Recovey	100%
Notifications	100%
Web Server Integration Tests	
API Gateway Routing	100%
Load Balancing Effectiveness	100%
Security Middleware Functionality	100%
Service Discovery and Registration	100%
Navigation Server Integration Tests	
Route request	100%
Data Parsing Accuracy	100%
Image Detection Server Integration Tests	
Integration with Camera Feeds	100%
Stress test on user requests to Image model	100%
Peripherals Server Integration Tests	
Sensor Data Aggregation	100%
Lid Mechanism Control	100%

Table 12. Integration Tests Summary

A.1.3. CI Pipeline showcase

Figure 4 shows a working CI pipeline using GitHub actions.

Figure 4. CI Pipeline Success Showcase

A.1.4. Google Lighthouse results

Figure 5 shows the Google Lighthouse scores on the web app.

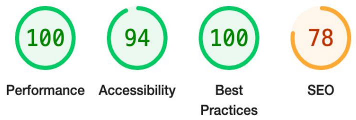

Figure 5. Google Lighthouse results

A.2. Sensor Results

A.2.1. Sensor fullness Readings

Distance from Sensor (cm)	Reading 1	Reading 2	Reading 3	Reading 4	Reading 5
5	100	100	100	100	100
10	66	66	66	66	66
15	66	66	66	66	66
20	66	66	66	33	66
25	33	33	33	33	66
30	33	33	33	33	33
35	0	33	33	33	33
40	0	33	0	0	33
45	0	0	0	0	0
50	0	0	0	0	0

Table 13. Sensor percentage fullness readings for distance

A.3. Waste Classification Training

A.3.1. Material Classification Training Loss

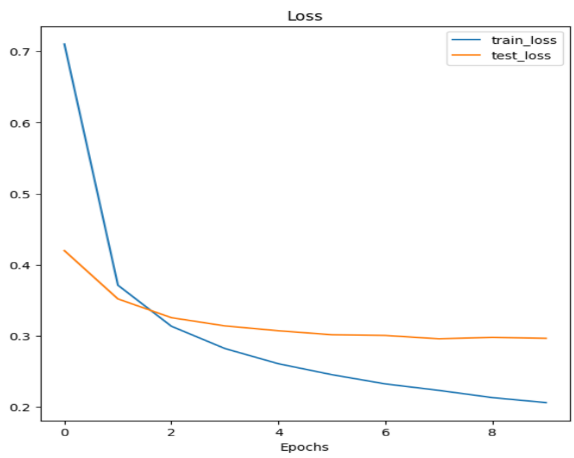

Figure 6. Training loss for the material classification model.

A.3.2. Material Classification Training Accuracy

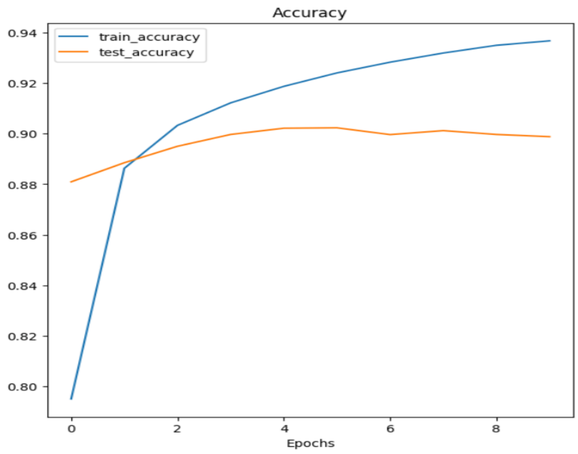

Figure 7. Training accuracy for the material classification model.

A.3.3. Cup Classification Training Loss

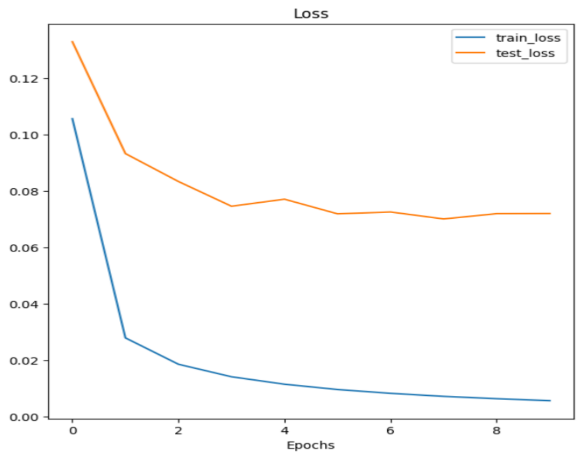

Figure 8. Training loss for the cup classification model.

A.3.4. Cup Classification Training Accuracy

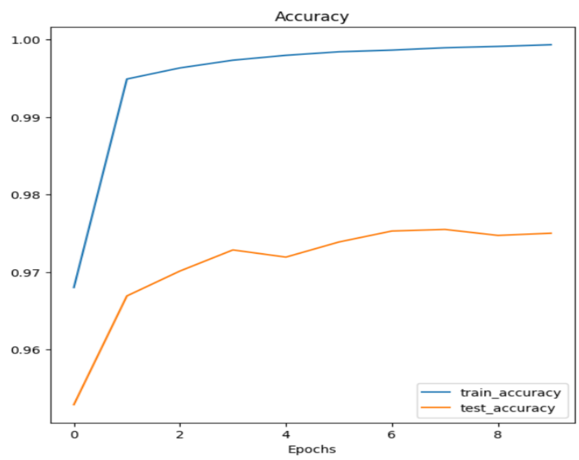

Figure 9. Training accuracy for the cup classification model.

A.3.5. Image Classification Model Architecture

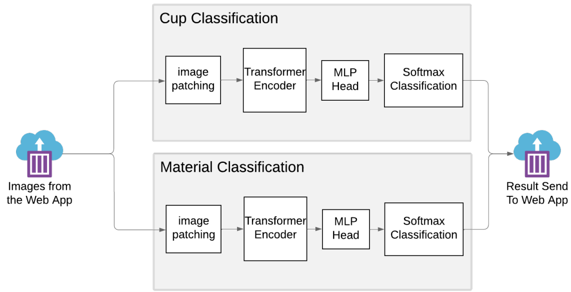

Figure 10. BinButler Image Model Architecture

A.3.6. Lid Response Testing

References

Dosovitskiy, Alexey, Beyer, Lucas, Kolesnikov, Alexander, Weissenborn, Dirk, Zhai, Xiaohua, Unterthiner, Thomas, Dehghani, Mostafa, Minderer, Matthias, Heigold, Georg, Gelly, Sylvain, Uszkoreit, Jakob, and Houlsby, Neil. An image is worth 16x16 words: Transformers for image recognition at scale. *CoRR*, abs/2010.11929, 2020. URL https://arxiv.org/abs/2010.11929.

Indeed. Product designer salary in united kingdom, 2024a. URL https://uk.indeed.com/career/product-designer/salaries.

Indeed. Software engineer salary in united kingdom, 2024b. URL https://uk.indeed.com/career/software-engineer/salaries.

Indeed. Robotics engineer salary in united kingdom, 2024c. URL https://uk.indeed.com/career/robotics-engineer/salaries.

Kumsetty, Nikhil Venkat, Bhat Nekkare, Amith, S., Sowmya Kamath, and Kumar M., Anand. Trashbox: Trash detection and classification using quantum transfer learning. In *2022 31st Conference of Open Innovations Association (FRUCT)*, pp. 125–130, 2022. doi: 10.23919/FRUCT54823.2022.9770922.

Majchrowska, Sylwia, Mikołajczyk, Agnieszka, Ferlin, Maria, Klawikowska, Zuzanna, Plantykow, Marta A., Kwasigroch, Arkadiusz, and Majek, Karol. Deep learning-based waste detection in natural and urban environments. *Waste Management*, 138:274–284, 2022. ISSN 0956-053X. doi: https://doi.org/10.1016/j.wasman.2021.12.001. URL https://www.sciencedirect.com/science/article/pii/S0956053X21006474.

Marionette. Plastic and paper cups synthetic image dataset, Aug 2022. URL https://www.kaggle.com/datasets/vencerlanz09/plastic-and-paper-cups-synthetic-image-dataset.

RS. Rs pro hybrid, permanent magnet stepper motor, 1.26nm torque..., 2024. URL https://uk.rs-online.com/web/p/stepper-motors/5350439?gb=s.

Sekar, Sashaank. Waste classification data, Jun 2019. URL https://www.kaggle.com/datasets/techsash/waste-classification-data.

Sproutsocial. Call to action (cta) - glossary, 2023. URL https://sproutsocial.com/glossary/call-to-action-cta/.