HORIZON-CL4-2021-DIGITAL-EMERGING-01
AI, Data and Robotics for the Green Deal (IA)

AI-powered Robotic Material Recovery in a Box

RECLAIM

D6.5: Algorithms and pipelines for Recycling Data Games

Contractual Date of Delivery:	31/02/2024
Actual Date of Delivery:	DD/MM/20YY
Security Class:	Public
Editor:	*Antonios Liapis (UM)*
Contributors:	Konstantinos Sfikas (UM), Nemanja Rasajski (UM), Georgios Alexakis (FORTH), Michail Maniadakis (FORTH)
Quality Assurance:	*Michail Maniadakis (FORTH)*
Deliverable Status:	Final

The *RECLAIM* Consortium

Part No.	Short Name of Participant	Participant Organization name	Country
1	FORTH	Foundation for Research and Technology Hellas	EL
2	UoM	University of Malta	MT
3	KUL	Katholieke Universiteit Leuven	BE
4	HERRCO	Hellenic Recovery Recycling Corporation	EL
5	IRIS	Iris Technology Solutions, Sociedad Limitada	SP
6	RBNS	ROBENSO PC	EL
7	AIMPLAS	AIMPLAS - Technological Institute of Plastics	SP
8	AXIA	Axia Innovation UG	DE
9	ISWA	International Solid Waste Association	NL
10	ION	Periferiakos Foreas Diaxirisis Stereon Apovliton Ionion Nison Anonimi Eteria Ton Ota	EL

Document Revisions

Version	Date	Editor	Overview
0.1	15/02/2024	Antonios Liapis (UM)	Structure and Early Draft
0.2	19/02/2024	Konstantinos Sfikas (UM), Nemanja Rasajski (UM)	Content additions on database format
0.3	11/03/2024	Antonios Liapis (UM)	Introduction, Future Work, Conclusion writeup
0.4	19/03/2024	Konstantinos Sfikas (UM)	Explanation of the final additions to games based on playtests
0.5	23/03/2024	Georgios Alexakis (FORTH), Michail Maniadakis (FORTH), Antonios Liapis (UM)	Addition of the standalone authoring tool and its details, reformatting of the document structure.
0.6	26/03/2024	Antonios Liapis (UM)	Final edits and typesetting

Table of Contents

List of Abbreviations

Abbreviation	Definition
AI	Artificial Intelligence
DoA	Description of the Action
RDG	Recycling Data Game
prMRF	portable, robotic Material Recovery Facilities
RoReWo	Robotic Recycling Workers
GDPR	General Data Protection Regulation
IP	Intellectual Property

Executive Summary

RECLAIM is a Horizon Europe funded project with an objective to **develop** a portable, robotic Material Recovery Facilities (MRFs) (**prMRF**) tailored to small-scale material recovery. RECLAIM adopts a modular multi-robot/multi-gripper approach for material recovery, based on low-cost Robotic Recycling Workers (RoReWos). An **AI module** combines imaging in the visual and infrared domain to identify, localize and **categorize recyclables**. Following a citizen science approach, RECLAIM aims to increase social sensitivity to the Green Deal as well as enhance its own AI algorithms with high-quality human annotations. Both goals (and more) are accomplished via a **novel Recycling Data-Game** that enables and encourages citizens to participate in project RTD activities by providing annotations to be used in **deep learning** for the **re-training of the AI module**.

This deliverable offers an update on the development of the recycling data game (RDG), following up on D6.2 submitted on M9. While D6.2 focused on early design and development efforts of mini-games that allow human players to annotate waste data in diverse scenarios (and with different degrees of cognitive challenge), D6.5 focuses more on the underlying infrastructure for data format, data collection and storage, and modelling. D6.5 does not re-iterate in depth the mini-games designed and developed so far: these were described in D6.2 and any changes made to the games were to integrate them with the underlying infrastructure and to fix minor technical issues. D6.5, instead, describes the database format and queries used to run the game, to collect human annotations, and to give rewards to players proportionate to the adherence of human annotations to some crowd-sourced ground truth. The deliverable also describes the data collected for awareness and education, and concludes with a view of the next steps for improving the engagement potential and intrinsic rewards of the RDG, given the current technical developments.

1. Introduction

RECLAIM proposes the development of a low cost, portable, easy to install and increased productivity prMRF that can achieve full material recovery anywhere, even in the most remote areas. The developed prMRF is expected to have a key role in developing a global, leakage-free circular economy model benefiting businesses, the society, and the environment.

However, we do not consider that a circular economy is only limited to material waste. With the complementary RECLAIM pillar (PIL-4) for *Environmental gaming for social awareness and data collection*, we envision that **data can also form a positive feedback loop and be re-used in a circular fashion**. Recycling data games (RDG) are proposed as a novel approach introduced by RECLAIM to enrich collected waste data with users' own feedback and thus improve the AI algorithms. In turn, better algorithms can filter which collected data is most ambiguous and thus relevant for users' feedback, achieving a self-sustaining (assuming user engagement) cycle of data re-use. The RDG developed under WP6 has a multitude of goals, including collecting user annotations (human data for the feedback loop), providing feedback to users on their contributions, increasing awareness through fact-sharing and question-giving, and promoting the results of the project.

This deliverable offers an update on the design and development of the recycling data game, complementing D6.2 submitted in M9. D6.2 presented the design of seven mini-games which primarily address the different human annotation tasks for data collection to address AI challenges for recyclable Identification, Categorization and Localization (AI-ILC). We summarise these seven mini-games in Section 2. This deliverable primarily concerns technological developments on the back-end, and infrastructure to store the user's data, use this data to calculate a crowdsourced "ground truth" and to reward players based on how well they match this ground truth. All of these functionalities are fundamental for the RDG (as data is at its core), and are described in Section 4. A custom database and API calls were developed (and described in Section 5). While this deliverable mostly concerns infrastructure additions, the data stored includes facts and questions intended for the player: we summarise these additions in Section 3. With this infrastructure in place, and the interfaces for data annotation tested under D6.3, we expect that future work will focus on integrating all the technical developments into a fun and engaging game: the roadmap described in Section 6 clarifies this. We conclude this deliverable in Section 7.

1.1 Intended readership

The present report is a public (PU) document. Its readership is considered to be the European Commission, the RECLAIM Project Officer, the partners involved in the RECLAIM Consortium, beneficiaries of other European funded projects, and the general public.

1.2 Relationship with other RECLAIM deliverables

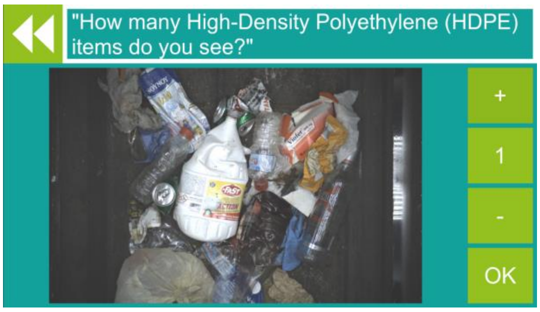
We note that the game design for the mini-games (see Section 2) is based on the user requirements collected under WP2 (specifically, D2.1). Since the current deliverable focuses on databases and infrastructure for deriving a "crowdsourced" ground truth, it is expected to directly impact AI algorithms that can use this ground truth for training. Thus, this deliverable is strongly linked to WP3 (Recyclable Waste Detection and Categorization). The image data used in this report was collected as part of D6.1 submitted in M9. Table 1 shows the main deliverables consulted (in case of past work), and impacted by (in case of future work) by this report.

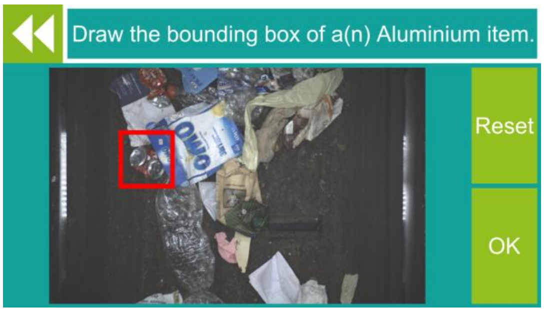
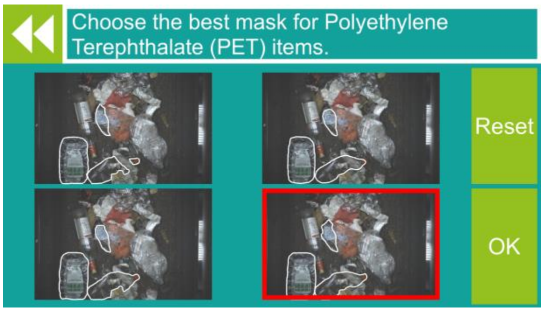
Del. No	Deliverable Name	WP	Month
1.1	Data management plan and ethics/privacy manual	WP 1	M6/M36
2.1	prMRF and RDG requirements and systems specification	WP 2	M6
3.1	Material recognition based on RGB and Hyperspectral imaging	WP 3	M18
3.2	prMRF operation monitoring and repeating advancement	WP 3	M30
6.1	Waste Data for material recognition and Recycling Data Game	WP 6	M9/M18
6.2	Algorithms and pipelines for Recycling Data Games	WP 6	M9
6.3	Assessment of the Recycling Data Game	WP 6	M18/M36
1.3	Final Project Report	WP 1	M36

Table 1: Other RECLAIM deliverables related.

2. Summary of Game Design

The Recycling Data Game will be a series of challenges for the player, thus offering short interactions that can be paused in-between challenges (based on user requirements collected in D2.1). In D6.2, and continuing in this deliverable, we focus on **annotation** challenges. However, **content knowledge** challenges and **content testing** challenges are also prepared (see Section 3) with a full knowledgebase for these types of challenges, stored in our database structure (see Section 5). Below, we describe the developed annotation challenges, each of which is referenced in Section 4 and Section 5 for the algorithms on ground truth detection and API calls respectively. Details about the games can be found in D6.2 and are omitted here for brevity. We include screenshots for each game based on the updated build of the game, which features functionality improvements and technical fixes found during internal testing.

Paint: the user must highlight all items of a specified material in each image using their finger (via a paintbrush and an eraser tool).	
Detect: the user must answer whether they can detect any item of a specific material in each image, using a Yes or No button.	
Count: the user must answer how many items of a specific material they can see in each picture, using a + and - button to increase/decrease the number.	

Outline: the user must choose one item of a specific material and draw a bounding rectangle around it, using their finger.	
Locate: the user must identify the center of a single item of a specific material using their finger (and a helper target graphic).	
Choose: the user is shown four different AI-generated masks around objects of a specific material, and must choose the best one among them.	
Categorize: the user is shown one AI-generated mask for all identified materials, and must choose which material each mask is via a "material" colour palette.	

2.1. Additions to the annotation challenges

The main additions to the challenges have been on the graphics side before and after the actual annotation challenge. We present all challenges' graphics changes below as they are very similar across challenges.

The introduction screen now offers more information about the challenge, including the material and the basic operation of the game.

Fig. 1: Intro Screen of the "Paint" challenge, explaining the basic operation of the game.

When the player has finished annotating an image, they press the Submit button, and submit their response to the server. The uploading process takes a few milliseconds, displaying a loading screen while in progress. After the user's response is submitted, it is compared against other players' responses to the same challenge, and a score is granted to the player accordingly, as shown in Figure 2. Detailed information regarding how the player's response is evaluated can be found in Section 4. This process is repeated for a sequence of 5 images. As soon as the session is complete, the player is shown a thank you message, as shown in Figure 3, explaining that the session is over and that their valuable input will be used to train better AI models.

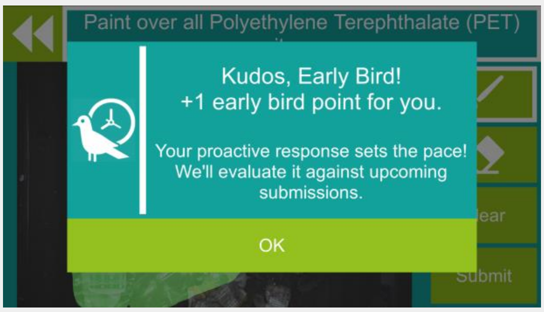


Fig. 2: The user's response is evaluated and the user is assigned a score.

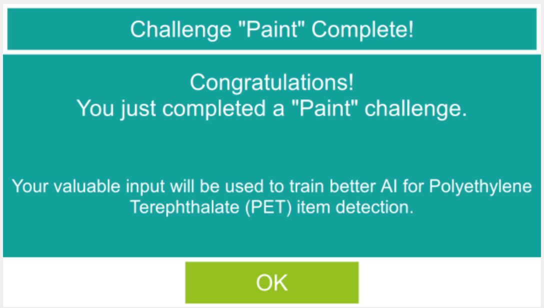


Fig. 3: Outro Screen of the "Paint" mini-game.

In addition, a player profile is added to the game. While this player profile will be enhanced in future work to increase the engagement (via gamification elements such as achievements etc., as discussed in Section 6), it serves an important purpose currently to inform the player about their rewards (points) in the different challenges. We revisit how points are calculated per challenge in Section 4. The player profile page is shown in Figure 4.

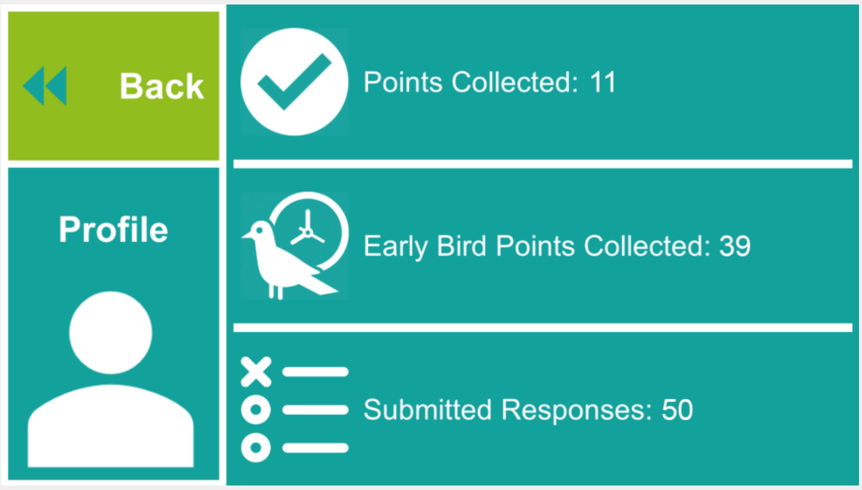


Fig. 4: Player profile after completing all challenges.

Since the current game is intended to be evaluated for usability as described in D6.3, some changes to the main interface were made. While in earlier versions of the RDG all mini-games were available on the start page, now the user must perform each challenge sequentially. The sequence of challenges is hand-crafted to ensure the easier challenges are first (e.g. detect, count) and on simpler images (e.g. images of isolated streams for PET bottles, as described in D6.1), increasing in difficulty and also moving to other materials (indicatively, glass, LDPE) and on mixed waste images where any challenge is more difficult. When the user exits the game, the system stores which challenge they were in. When the player re-opens the RECLAIM RDG, the system restores the player's local load file (their last completed challenge) and checks their player profile from the server, thus letting the player continue where they left off. In total, 10 challenges are implemented in the following order:

1. **"Count"** PET items (data: isolated stream with PET Only)
2. **"Locate"** PET items (data: isolated stream with PET Only)
3. **"Outline"** PET items (data: isolated stream with PET Only)
4. **"Detect"** HDPE items (data: Mixed Materials stream)
5. **"Count"** GLASS items (data: Mixed Materials stream)
6. **"Locate"** HDPE items (data: Mixed Materials stream)
7. **"Outline"** GLASS items (data: Mixed Materials stream)
8. **"Choose"** PET items (data: Mixed Materials stream)
9. **"Paint"** PET items (data: Mixed Materials stream)
10. **"Categorize"** all materials (data: Mixed Materials stream)

3. Knowledge base for Content Testing Challenges

As identified in the initial game design (D6.2), awareness around the impact of recycling will be achieved via **content knowledge** challenges and **content testing** challenges. In the refined game design, we tried to move beyond the “Information Deficit Model” for behavioural change through serious games [Tanenbaum2013] to pair these challenges as **content testing** challenges. The questions for these challenges are linked to authentic experiences of every-day life (e.g. ways to reduce food waste or reuse items). The user is first asked a question and, after answering, is presented with elaborate feedback on the facts relating to this question. This approach is expected to increase interest, curiosity, retrieval practices and retention to memory since the user is more invested in knowing the facts relating to the questions asked [Zeglen2018] [Roediger2011].

Content testing challenges, therefore, are designed to broaden the players' understanding of various recycling-related topics. In the database (see Table 1) facts and questions are grouped into themes. Themes in the current version, include: (1) information about different policies, (2) impact of recycling, (3) sorting facilities management, (4) landfills management, (5) information about different types of materials and more. To accommodate a wider audience and facilitate bilingual gameplay, the entire questionnaire is available in both Greek and English.

From a technical standpoint, the knowledge base consists of 135 recycling-related questions, derived from 110 facts (multiple questions can be based on a single fact). The available responses can be binary (True/False) or multiple-choice. While the majority of these questions are objective, with clear correct answers, a few of them are subjective, designed to provoke thought without a definitive right or wrong answer. After submitting their answers, players receive concise yet informative paragraphs that shed light on broader aspects of the topic at hand. To further aid understanding, some facts are enhanced with accompanying images or graphics, visually illustrating the information provided. Table 1 summarises various details regarding the available questionnaire data.

Table 1: Statistics of the current knowledge base for Content Testing challenges

Number of Facts	110
Number of Questions	135
Number of questions with 2 options	81

Number of “True / False” questions	70
Number of “Yes / No” questions	4
Number of questions with a single correct answer	131
Number of questions with multiple correct answers	2
Number of questions with no correct answers	2

3.1 Collection Process

Themes, such as the interest of players on the impact of recycling, emerged from focus groups discussions with 16 participants in total, which took place in January 2023 and were reported under D2.2.

Facts and questions regarding waste management, recycling, and relevant environmental issues, were collected via collaboration of a team of experts from the partner institutions. Specifically, ISWA provided information, material and resources relevant to waste management, recycling and implications. HERRCO provided general information about recycling and waste management, information of local interest (Greece), and insights on the target group needs and requirements. Involvement of experts focused on waste management and recycling ensures validity and authenticity of the game content.

Certainly, other valid resources are used for the environmental content of the game such as publicly available information at the Hellenic Recycling Agency (https://grecycle.gr/, https://www.eoan.gr/), and the Greek Ministry of Environment and Energy (https://ypen.gov.gr).

3.2 Data format of the Question-Answering challenges

The Question-Answering challenges are currently stored in the database, but not yet integrated with the RECLAIM Data Collection Game. Their organisation and structure, however, is such that it can directly support an updated version of the game that will utilise them. The relevant data are stored in the database in two tables, named QA_Facts and QA_Questions, as explained in the following paragraphs.

All facts are stored in a database (DB) table named **QA_Facts**. A fact is a piece of information (relevant to the theme of recycling), accompanied by meta-information, such as a theme and potentially a relevant image. The precise structure of the QA_Facts table is presented in the following list, where each bullet represents a column of the table:

- **Fact ID** (TEXT)
  A unique Identifier for this fact.
- **Theme** (TEXT)
  A tag for this fact, out of the following list: [“Info about target materials”, “Less waste in landfills”, “Reduce”, “Reuse”, “Recycle”, “What happens in a sorting facility?”, “What happens after the sorting/recycling facility?”, “Policies”, “What is the impact of recycling?”, “Other”].
- **Fact** (TEXT)
  A json file including the fact in the two available languages: Greek and English.
- **Image Data** (TEXT)
  An image related to the fact.
- **Image Source Link** (TEXT)
  A link to the source of the image, if retrieved online (e.g., wikipedia)
- **Image Info** (TEXT)
  Extra information regarding the image, such as licence information or other.

A single fact can be the basis for posing many questions. On the contrary, each provided question is based on a single fact. Based on this game-design choice, we keep the questions in a separate table, named **QA_Questions**, to avoid data repetition and reduce maintenance workload. The precise structure of the QA_Questions table is presented in the following list, where each bullet represents a table column:

- **Question ID** (TEXT): A unique identifier of this question.
- **Fact ID** (TEXT): Reference to a fact upon which the question is based.
- **Question** (TEXT): A Json file including the question in Greek and English.
- **Options** (TEXT): A Json file including the available answers in Greek and English
- **Correct Answer Index** (INT): The index of the correct answer (or -1 if the answer is subjective).
- **Notes** (TEXT): Extra information about the question, such as who proposed it, etc.

4. Ground truth detection and game scoring

A core challenge with citizen science games, and the RDG in particular, is that user data is collected in lieu of expert annotations. Therefore, there is no ground truth on what a correct user response is. Each mini-game, as outlined in Section 2, is crafted to yield specific types of user-generated data, pivotal for training or retraining specialised AI models for object detection. Based on current practices in crowdsourced data collection tasks (such as CAPTCHA) and the particularities of the RDG datasets and mini-games developed so far [Arnab2019; Strobl2019; Sumner2020], a methodology has been devised for validating collected user data.
An important aspect for the gamification of the RDG is rewarding the player for their annotation tasks. Since no definitive ground truth exists for correct annotations, a corpus of "definitive" user annotations is essential (we refer to this as the “ground truth”). Thus, preliminary user annotations within a specific mini-game for a specific image are rewarded a participation incentive known as the "early bird" reward until reaching a predetermined minimum quota or "scoring threshold." In cases of strong disagreements among initial annotations, such as in the "Detect" mini-game where 50% of users claim True while 50% claim False, the "early bird" award persists until sufficient annotator agreement surpasses the "agreement threshold." These thresholds are tailored per game, adjustable to accommodate varying user quantities and levels of annotation certainty, with default values set for the minimum annotations of the same data point (“scoring threshold”) to **10** and for the "agreement threshold" to **50%**.

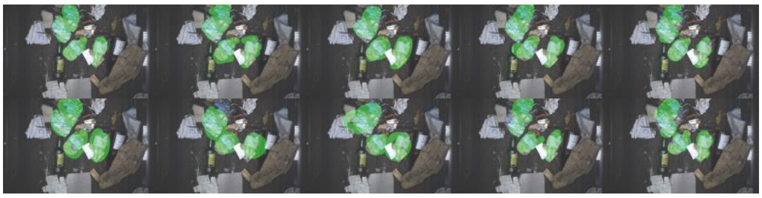
4.1 Current implementation

The following paragraphs outline the types of data collected from each of the developed mini-games, the methods for establishing ground truths, allocating points to users, and how this data can be utilised for machine learning.

4.1.1 “Paint” mini-game

As described in Section 2, the “Paint” mini-game asks the user to highlight the regions of an image that include items of a specific material. Every time the user annotates a reference image in this manner, they essentially generate a binary mask, in the form of a monochrome bitmap (as seen in Fig. 5). This representation enables us to adopt a straightforward method for detecting ground truth and allocating points. To calculate agreement we aggregate all previous responses, which gives us a count of how frequently each pixel in the images has been painted over. We then use the convenient property of pixel maps, where new responses can be utilised to extract pixels from the previously accumulated sum. Our similarity metric is finally created by summing the total occurrences of each pixel being selected, then dividing by the number of pixels painted over in the new response. This provides a measure of how closely, on average, the new response aligns with previous ones, a visual representation of this method can be seen on Fig. 6. The "agreement threshold" signifies the average proximity of pixels in a new response to those in previous user responses. If the previously computed value exceeds the "agreement threshold", full points are awarded; otherwise, no points are given.

(1) Reference image

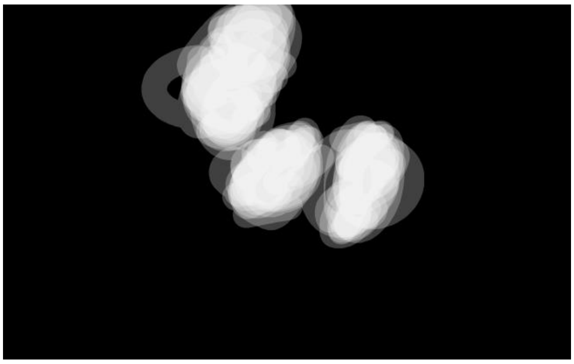


(2) Annotations of PET bottles from 10 different users, for the same reference image.

(3) Binary classification masks (per pixel), generated from the user annotations.

Fig. 5: A reference image (1), where the user is asked to highlight all PET bottles. In (2), a number of annotations from different users and in (3), their generated binary masks.

(1) Reference image, where the users are asked to highlight all PET bottles.



(2) Averaged user annotations, resulting in a probabilistic classification “heatmap”.

Fig. 6: Overlaid (averaged) user input masks collected via the “Paint” mini-game. Such overlaid data can be used to estimate the average probability for the existence of items of a specific material, within the image.

4.1.2 “Detect” mini-game

In the “Detect” mini-game the user is asked whether they can detect any items (one or more) of a specified material in a reference image. In doing so, they provide a binary (True / False) annotation for the reference image.
After a corpus of user annotations per image per mini-game is collected, then new user annotations can be awarded based on how close they are to the consensus. For simple classification tasks like this determining agreement is trivial:

- If the user selects the most popular class (e.g. True or False for PET object in image) and this response is above “agreement threshold” they are awarded full points, otherwise they do not receive any points.

4.1.3 “Count” mini-game

In the “Count” mini-game, the user is asked how many items of a specified material they can spot in a reference image. Their numeric answer is a form of multi-class annotation for the reference image, where the available classes are 0 to positive infinity.
After a corpus of user annotations per image per mini-game is collected, then new user annotations can be awarded based on how close they are to the consensus. For simple classification tasks like this determining agreement is trivial:

- If the user selects the most popular class (e.g. user detects 2 PET objects in image) and this response is above “agreement threshold” they are awarded full points, otherwise they do not receive any points.

4.1.4 “Outline” mini-game

In the “Outline” mini-game, the user is asked to draw a bounding rectangle, surrounding an item of a specified material on a reference image. In doing so, they define a region in the image, within which an item of the specified material is supposed to exist.
To address the problem of ground truth detection, we apply clustering, where we calculate the distance between points using Intersection over Union (IoU). IoU describes the extent of overlap of two boxes and its values range between 0 and 1. A higher IoU value indicates better overlap between two boxes. To use IoU as a distance metric it needs to be inverted, so distance between two points is calculated as 1 - IoU. Since the number of ground truths isn’t known beforehand, we apply the Density-Based Spatial Clustering of Applications with Noise (DBSCAN) algorithm. This algorithm can find clusters of points that are close to each other within a certain distance threshold. The "agreement threshold" defines the minimum distance required for two bounding boxes to be considered part of the same cluster. If the new response is within a cluster we award full points, otherwise we award no points.

(1) Reference Image

(2) 10 different user inputs, detecting PET items in the reference image, acquired via the RDG

Fig. 7: (1) A reference image where the user is asked to spot PET items. (2) Different user annotations, as they appear within the RDG.

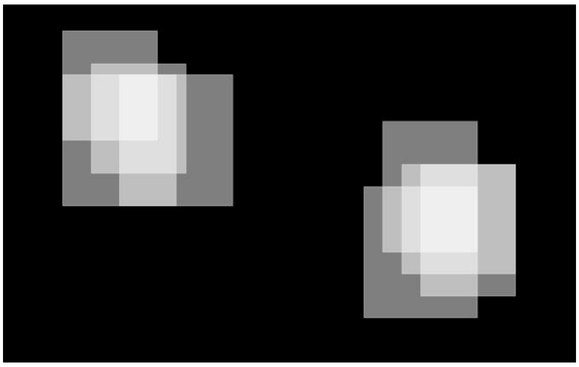


Fig. 8: Visual representation of two clusters with IoU distance overlap.

4.1.5 “Locate” mini-game

In the “Locate” mini-game, the user is asked to indicate the centre of an item of a specified item, on a reference image. Their input is stored as a single 2D point, representing the image-coordinates where the user clicked (Fig. 9).

Ground truth detection, and thereby scoring, depends on how closely a new user's response matches the densely populated areas of an image, found using kernel density estimation (KDE). The "agreement threshold" serves as a proximity metric of the new response to areas of maximal density within the image. The process involves constructing a Gaussian kernel density model using previous responses for calibration. Subsequently, the log-likelihood of the new response is computed and normalised by the highest extant density value. Points are allocated based on a comparison between this normalised likelihood and the "agreement threshold." In essence, the scoring mechanism rewards users whose responses closely approximate the densest segments of the image. Fig. 10 represents a visual representation of this process.

Fig. 9: Reference Image, where the users are asked to indicate the centre of PET items, and indicative input of multiple users, as shown in the RDG.

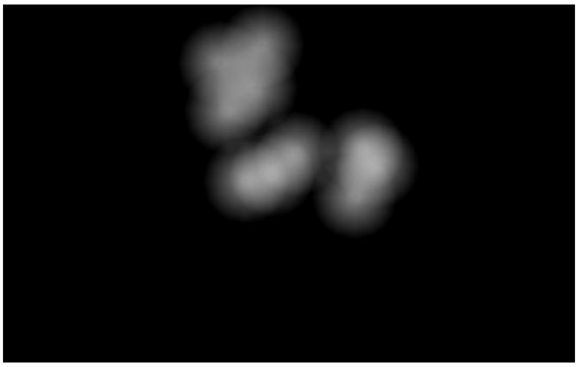


Fig. 10: KDE of the point-cloud, representing the probability distribution of PET items in the image.

4.1.6 "Choose" mini-game

In the "Choose" mini-game, the user is asked to compare a set of pre-computed masks, and select the one that better represents the regions of a reference image that include a specific material. The presented masks represent the boundaries of all items of a specific material, within the reference image, and the user's comparison criterion has to do with the correctness and precision of presented boundaries.

After a corpus of user annotations per image per mini-game is collected, then new user annotations can be awarded based on how close they are to the consensus. For simple classification tasks like this determining agreement is trivial:

- If the user selects the most popular class (e.g. the second pre-computed mask matches all PET objects in image) and this response is above "agreement threshold" they are awarded full points, otherwise they do not receive any points.

4.1.7 “Categorize” mini-game

In the “Categorize” mini-game, the user is presented with a single image, and within it, the boundaries of a number of objects. Then, they are asked to classify each one of these presented regions as belonging to one of a set of possible materials. In contrast to the previous challenges, here the emphasis is not on the precision of each boundary, but rather on its correct classification into the proper material.
We determine ground truths for each region separately, by applying above mentioned simple classification logic region-wise (e.g. if there are 5 boundaries we assign 5 different scores depending on how close each boundary is to the consensus). To be consistent with the reward assignment process in other mini-games we aggregate the reward into a single value using the following logic:

- If “scoring threshold” is not met for all boundaries, we assign “early bird” points
- If the user response was correct for all boundaries we assign all points
- If the user response consists of correct and “early bird” points (e.g. for 6 boundaries we have 3 correct and 3 “early bird” rewards, or 2 correct, 2 “early bird” and 2 incorrect) we assign all points.
- In any other combination the user does not receive points.

Both detailed and aggregated rewards are persisted in the database.

4.2 Future considerations

In the next phase of the project, depending on feedback received from user tests (under T6.3), more flexible reward assignment with partial points will be considered. Users would be rewarded based on the prevalence of their preferences among other users. Each annotation would still receive a maximum allocation of points, under the assumption of unanimous voting. Adjustments for partial points would be made proportionally to the deviation from 100% agreement among users. For instance, in a scenario where four AI-generated masks are presented in a "Choose" minigame, if Mask A has been selected by 10% of users, Mask B by 20%, Mask C by 75%, and Mask D by 5%, selecting Mask B might yield 20% of the maximum resources, whereas selecting Mask C could result in 75% of the maximum resources. Further testing and refinement of resource allocation strategies based on user preferences will be conducted in subsequent iterations. This approach aims to incentivize continued engagement through participation rewards while also encouraging users to provide *accurate* annotations.

It is worth noting that this will not be the only reward provided to players, as establishing the ground truth (while critical to the AI algorithms) is of lesser interest to the players (see CSG players’ motivations in D2.1). Therefore, other resources will reward continued engagement and participation in order to motivate the player to keep engaging with the RDG. Therefore, incentives to improve the users’ performance in annotation tasks will be orthogonal to other incentives to continue playing the RDG and participating in the research effort longitudinally. Details of the reward structure based on ratio of agreement (and how lenient or harsh this may be) will be refined with preliminary user tests and take into account recommendations from the literature. Indicatively, [[See2016]~~Foody2014]~~ warns that *"introducing overly burdensome structures to ensure quality could damage the potential contributions from related socially-conscious and citizen-focused data collection and mapping efforts."*

5. RECLAIM API

Reclaim API is the back-end created to support the functionality of and data persistence of user responses of above introduced mini-games. It is implemented in Python 3.8 using Flask framework and is available at: https://reclaimgame.institutedigitalgames.com. It currently supports the following functionality:

- Initialization of database
- Batch population of images (images need to be provided beforehand)
- Batch population of detection processes necessary for “Choose” and “Categorize” mini-games (detection processes need to be provided beforehand in .json format)
- Registration of new users
- Login for existing users
- Randomised data acquisition (which includes images and materials that need to be annotated) for "Detect", "Count", "Choose", “Categorize”, “Paint”, “Outline” and “Locate” mini-games
- Acquisition of historical user data which includes images they’ve annotated and total points they’ve acquired
- Insertion of images and AI Process JSON files
- Insertion of new user response entries for "Detect", "Count", "Choose", “Categorize”, “Paint”, “Outline” and “Locate” min-igames
- Reward assignment for new user response entries for "Detect", "Count", "Choose", “Categorize”, “Paint”, “Outline” and “Locate” mini-games.

5.1 Data structure

The Reclaim API stores all necessary data in a relational MySQL database, which can be roughly categorised into two segments. The first segment includes essential game and user data needed for basic functionalities such as creating mini-game instances and managing users. The second segment comprises tables dedicated to storing user responses for various minigames like "Detect," "Count," "Choose," "Categorize," "Paint," "Outline," and "Locate," along with their corresponding calculated scores.

5.1.1 Supporting data structure

The game data and user profile tables store images for annotation, machine-generated masks, and user profile data. This section of the database comprises the following tables, with a detailed overview depicted in Fig. 11:

- “materials_description” - this table stores data about existing materials that will be annotated though mini-games

- “registered_device_ids” - currently this table stores identifiers of devices that mini-games are being played on, along with assigned user identifiers
- “ai_detection_process” - this table store information about detection process files for images persisted in the database
- “ai_detected_objects” - this table stores machine generated masks for each detected object along with necessary details
- “conveyor_belt_image” - this table contains images encoded into base64 string, along with image metadata (e.g. aspect ratio, height, width, format)

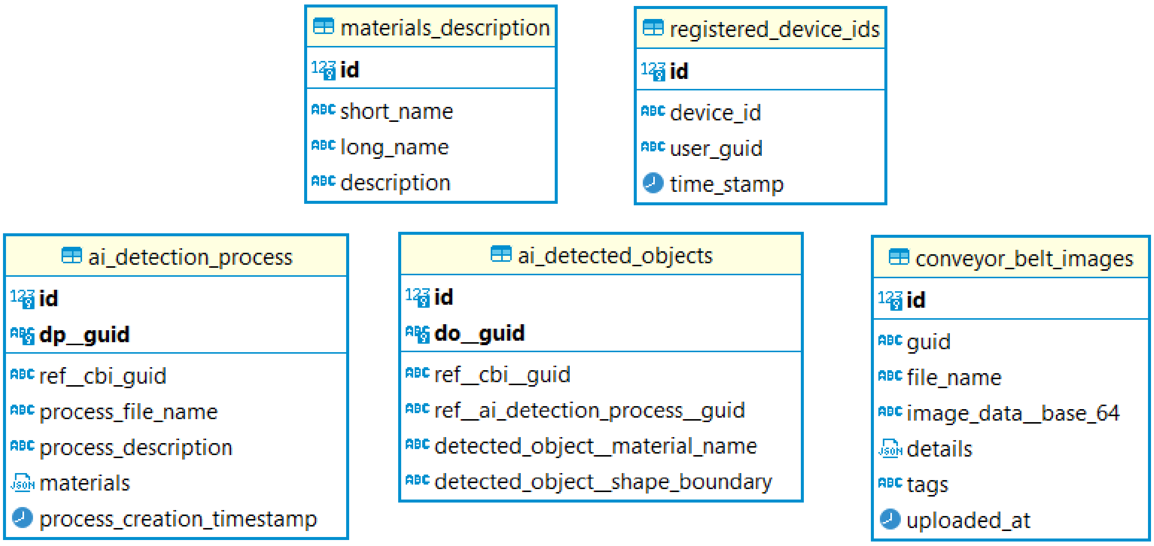


Fig. 11: ER diagram of game and user data tables.

5.1.2 User response data structure

The user response tables serve to store user annotations for various mini-games such as "Detect," "Count," "Choose," "Categorize," "Paint," "Outline," and "Locate", along with details about the annotated images, user identifiers, and awarded points as stated in Section 4. This section of the database encompasses the following tables, presented with a comprehensive overview in Fig. 12:

- “user_responses__detect” - this table stores response for the “Detect” mini-game, which include raw annotation in the form of a binary answer (1 for True, 0 for False)
- “user_responses__count” - this table stores response for the “Count” mini-game, which include raw annotation in the form of number of detected objects (integer value)
- “user_responses__choose” - this table stores response for the “Choose” mini-game, which include detection processes that were shown to the user and selected process

- “user_responses__categorize” - this table stores response for the “Categorize” mini-game, which include detection process that user was shown, along with detailed and aggregates awarded points
- “user_responses__locate” - this table stores response for the “Locate” mini-game, which include raw annotation in the form of x and y coordinates of selected pixel
- “user_responses__outline” - this table stores response for the “Outline” mini-game, which include raw annotation in the form of bottom left, and top right bounding box coordinates
- “user_responses__paint” - this table stores response for the “Count” mini-game, which include raw annotation in the form of binary pixel map

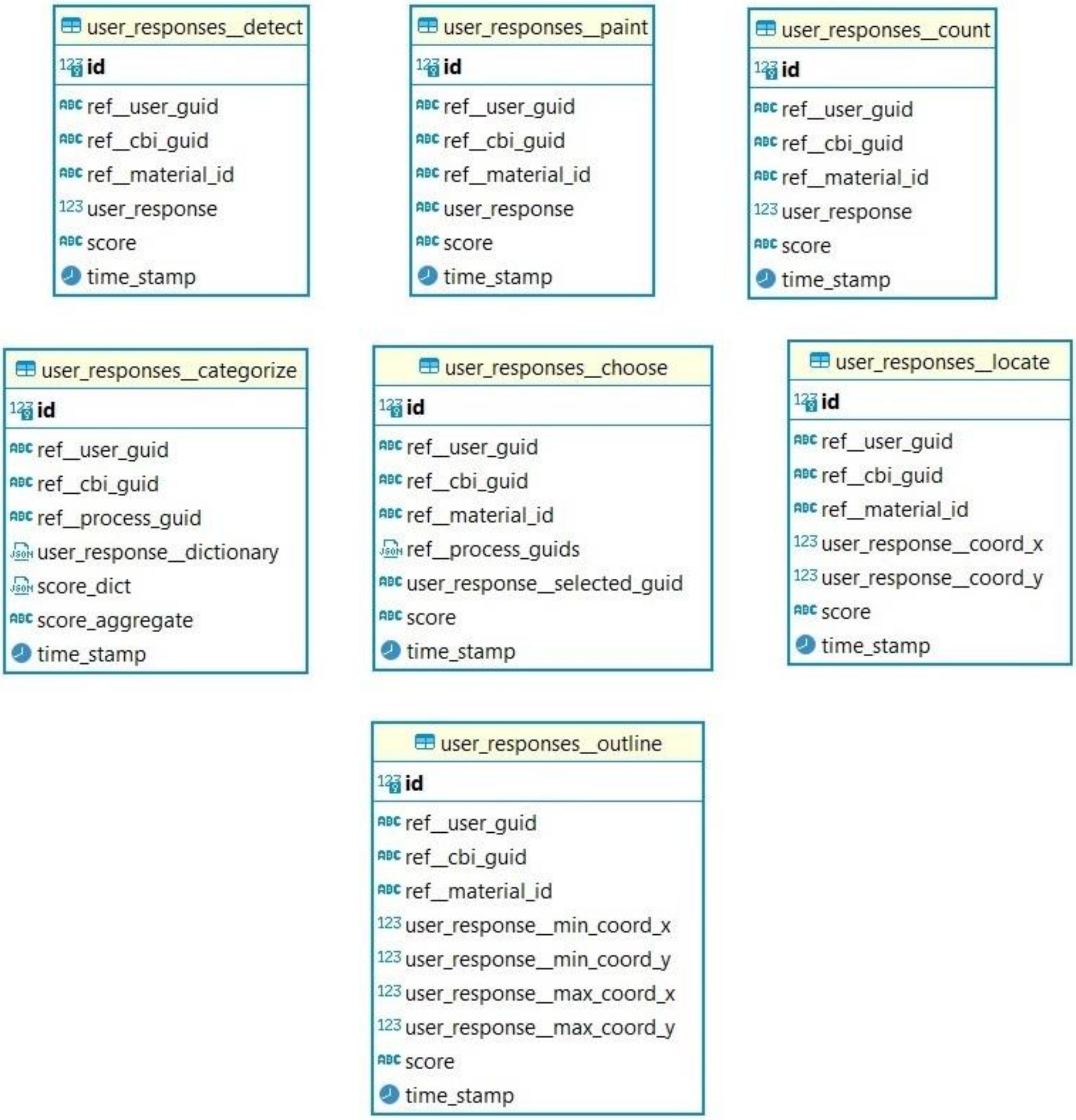


Fig. 12: ER diagram of user response tables.

5.2 API endpoints

5.2.1 Create tables

Creates all tables necessary for the operation of Reclaim API.

Request

Method	URL
POST	/create_tables

Response

Status	Response
200	[["ai_detected_objects"], ...]

5.2.2 Drop tables

Deletes all tables in the database related to Reclaim API.

Request

Method	URL
POST	/drop_tables

Response

Status	Response
200	[]

5.2.3 Populate base data

Inserts all images and detection processes that are uploaded to dedicated folders inside Reclaim API filesystem.

Request

Method	URL
POST	**/populate_base_data**

Response

Status	Response

200	{
	 "ai_objects": [
	 [
	 NUMBER_OF_INSERTIONS
	]
	],
	 ...
	 }

5.2.4 Populate synthetic data

Inserts 1000 randomly generated user profile entries, and user responses for "Detect", "Count", "Choose", “Categorize”, “Paint”, “Outline” and “Locate” mini-games. These are used for internal testing of e.g. ground truth calculation algorithms (see Section 4).

Request

Method	URL
POST	/populate_synthetic_data

Response

Status	Response
200	{ "categorize": [[NUMBER_OF_INSERTIONS]], ... }

5.2.5 User game history

Returns history of played games and images that were used in those games for specified user.

Request

Method	URL
GET	/user_game_history

Request Parameters

Type	Params	Values
URL_PARAM	user_id	string

Response

Status	Response

200	[
	  {
	    "game": "detect",
	    "mix": NUMBER_OF_ANNOTATIONS
	  },
	 …
	]

5.2.6 Total points

Returns total points for specified user.

Request

Method	URL
GET	/total_points

Request Parameters

Type	Params	Values
URL_PARAM	user_id	string

Response

Status	Response
200	{ "EARLY_BIRD": N1, "RIGHT": N2, "WRONG": N3 }

5.2.7 Login

Creates new user entry if one does not exist, or returns *user_id* if entry exists.

Request

Method	URL
POST	/login_or_register

Request Parameters

Type	Params	Values
POST_DATA	device_id	string

Response

Status	Response
200	"USER_ID_UUID_V4"

5.2.8 Insert images

Inserts a new (unique) image into the database. File name is inspected to verify image uniqueness. This is an important component for integration of the live data collection described in D6.4 (Waste Data for material recognition and Recycling Data Game) submitted concurrently.

Request

Method	URL
POST	/insert_image

Request Parameters

Type	Params	Values
POST_DATA	file_name	string
POST_DATA	image_data	string - base64 serialised image
POST_DATA	details	string - (serialised JSON object with height, width and file type)
POST_DATA	tags	string - [“only_pet”, “mix”]

Response

Status	Response
201	Ok
400	Client request error
409	Image already exists in database

5.2.9 Insert AI Process JSON files

Inserts a new (unique) AI Process into the database. File name is inspected to verify AI process uniqueness.

Request

Method	URL
POST	/insert_ai_process

Request Parameters

Type	Params	Values
POST_DATA POST_DATA	file_name file_data	string string - (serialised JSON object)

Response

Status	Response
201	Ok
400	Client request error
409	AI Process already exists in database
412	AI Process references image that does not exist in database

5.2.10 Paint game

Endpoints for fetching randomly selected images and material, award assignment and insertion of new entries for "Paint" game.

5.2.10.1 GET game data

Request

Method	URL
GET	/paint

Request Parameters

Type	Params	Values
URL_PARAM URL_PARAM URL_PARAM	num_games tag response_type	int - [1,...,N) string - ["only_pet", "mix"] string - ["as_object", "as_list"]

Response

Status	Response

200	[
	{
	"image_data": "...",
	"image_id": "...",
	"material": "..."
	}
	]

5.2.10.2 Game scoring and insertion into database

Request

Method	URL
POST	/paint

Request Parameters

Type	Params	Values
POST_DATA	img_id	string
POST_DATA	user_id	string
POST_DATA	material_id	string
POST_DATA	user_response	string

Response

Status	Response
200	{ "game_result": "", "insert_response": "1 records inserted successfully into table" }

5.2.11 Choose game

Endpoints for fetching randomly selected images and material, award assignment and insertion of new entries for “Choose” game.

5.2.11.1 GET game data

Request

Method	URL
GET	/choose

Request Parameters

Type	Params	Values
URL_PARAM URL_PARAM URL_PARAM	num_games tag response_type	int - [1,...,N) string - [“only_pet”, “mix”] string - [“as_object”, “as_list”]

Response

Status	Response

200	[
	{
	"image_data": "...",
	"image_id": "...",
	"material": "..."
	}
	]

5.2.11.2 Game scoring and insertion into database

Request

Method	URL
POST	/choose

Request Parameters

Type	Params	Values
POST_DATA	img_id	string
POST_DATA	user_id	string
POST_DATA	material_id	string
POST_DATA	ai_process_ids	string (serialised JSON array)
POST_DATA	user_selected_ai_process_id	string

Response

Status	Response

200	{
	 "game_result": "",
	 "insert_response": "1 records inserted successfully into table"
	 }

5.2.12 Categorize game

Endpoints for fetching randomly selected images and material, award assignment and insertion of new entries for "Categorize" game.

5.2.12.1 GET game data

Request

Method	URL
GET	/categorize

Request Parameters

Type	Params	Values
URL_PARAM URL_PARAM URL_PARAM	num_games tag response_type	int - [1,...,N) string - ["only_pet", "mix"] string - ["as_object", "as_list"]

Response

Status	Response

200	[
	{
	"image_data": "...",
	"image_id": "...",
	"material": "..."
	}
	]

5.2.12.2 Game scoring and insertion into database

Request

Method	URL
POST	/categorize

Request Parameters

Type	Params	Values
POST_DATA	img_id	string
POST_DATA	user_id	string
POST_DATA	ai_process_id	string
POST_DATA	user_response_json	string (serialised JSON object)

Response

Status	Response
200	{ "game_result": "", "insert_response": "1 records inserted successfully into table" }

5.2.13 Outline game

Endpoints for fetching randomly selected images and material, award assignment and insertion of new entries for “Outline” game.

5.2.13.1 GET game data

Request

Method	URL
GET	/outline

Request Parameters

Type	Params	Values
URL_PARAM URL_PARAM URL_PARAM	num_games tag response_type	int - [1,...,N) string - [“only_pet”, “mix”] string - [“as_object”, “as_list”]

Response

Status	Response

200	[
	{
	"image_data": "...",
	"image_id": "...",
	"material": "..."
	}
	]

5.2.13.2 Game scoring and insertion into database

Request

Method	URL
POST	/outline

Request Parameters

Type	Params	Values
POST_DATA	img_id	string
POST_DATA	user_id	string
POST_DATA	material_id	string
POST_DATA	x_min	int [-1,..., IMAGE_WIDTH]
POST_DATA	y_min	int [-1,..., IMAGE_WIDTH]
POST_DATA	x_max	int [-1,..., IMAGE_WIDTH]
POST_DATA	y_max	int [-1,..., IMAGE_WIDTH]

Response

Status	Response
200	{ "game_result": "", "insert_response": "1 records inserted successfully into table" }

5.2.14 Locate game

Endpoints for fetching randomly selected images and material, award assignment and insertion of new entries for "Locate" game.

5.2.14.1 GET game data

Request

Method	URL
GET	/locate

Request Parameters

Type	Params	Values
URL_PARAM URL_PARAM URL_PARAM	num_games tag response_type	int - [1,...,N) string - ["only_pet", "mix"] string - ["as_object", "as_list"]

Response

Status	Response

200	[
	{
	"image_data": "...",
	"image_id": "...",
	"material": "..."
	}
	]

5.2.14.2 Game scoring and insertion into database

Request

Method	URL
POST	/locate

Request Parameters

Type	Params	Values
POST_DATA POST_DATA POST_DATA POST_DATA POST_DATA	img_id user_id material_id x y	string string string int [-1,..., IMAGE_WIDTH] int - [-1,..., IMAGE_HEIGHT]

Response

Status	Response

200	{
	"game_result": "",
	"insert_response": "1 records inserted successfully into table"
	}

5.2.15 Count game

Endpoints for fetching randomly selected images and material, award assignment and insertion of new entries for “Count” game.

5.2.15.1 GET game data

Request

Method	URL
GET	/count

Request Parameters

Type	Params	Values
URL_PARAM URL_PARAM URL_PARAM	num_games tag response_type	int - [1,...,N) string - [“only_pet”, “mix”] string - [“as_object”, “as_list”]

Response

Status	Response

200	[
	{
	"image_data": "...",
	"image_id": "...",
	"material": "..."
	}
	]

5.2.15.2 Game scoring and insertion into database

Request

Method	URL
POST	/count

Request Parameters

Type	Params	Values
POST_DATA	img_id	string
POST_DATA	user_id	string
POST_DATA	material_id	string
POST_DATA	user_response	int - [0,...,N)

Response

Status	Response
200	{ "game_result": "", "insert_response": "1 records inserted successfully into table" }

5.2.16 Detect game

Endpoints for fetching randomly selected images and material, award assignment and insertion of new entries for “Detect” game.

5.2.16.1 GET game data

Request

Method	URL
GET	/detect

Request Parameters

Type	Params	Values
URL_PARAM	num_games	int - [1,…,N)
URL_PARAM	tag	string - [“only_pet”, “mix”]
URL_PARAM	response_type	string - [“as_object”, “as_list”]

Response

Status	Response

200	[
	{
	"image_data": "...",
	"image_id": "...",
	"material": "..."
	}
	]

5.2.16.2 Game scoring and insertion into database

Request

Method	URL
POST	/detect

Request Parameters

Type	Params	Values
POST_DATA	img_id	string
POST_DATA	user_id	string
POST_DATA	material_id	string
POST_DATA	user_response	int - [0, 1]

Response

Status	Response
200	{ "game_result": "", "insert_response": "1 records inserted successfully into table" }

6. Companion online application for in-depth annotation

In addition to the RDG mobile application, which offers human computation on tasks for Identification, Categorization and Localization, we acknowledge that RDG challenges allow for high-level feedback (such as number or presence of specific materials, rather than the exact boundaries of each). Concerns about mobile phone screens and tactile drawing, as well as issues of time commitments raised during focus groups of D2.2, drove these design decisions for the RDG in order to maximise the number and value of human annotations without overburdening them. However, for interested users that wish to move beyond the gamified interactions of the RDG, a companion application was developed by FORTH to annotate in a more complex and controlled way the same data offered by the RDG.

6.1 AI Annotation Tool

In the field of deep learning for object recognition in RGB images, the use of fast annotation tools may crucially support the training and optimization of the AI models. The development of an efficient image annotation tool is necessary as it allows fast and accurate labelling of objects to save annotation time, and definition of segmentation boundaries between adjacent objects that form the basis for training models to distinguish between objects of different types. Besides accelerating the processing of raw data, an efficient annotation tool can also facilitate the rapid generation of large-scale synthetic datasets and reduce the time required to develop robust object identification, classification and localization models. Additionally, a well-designed annotation tool ensures consistency across datasets, reliably promoting the replication of results, while its user-friendly interface enhances accessibility and collaboration within the RECLAIM consortium.

To facilitate the annotation of waste images, FORTH has developed a new solution to image annotation, the so called "Coffee Break", which leverages open source technologies and integrates available artificial intelligence models as external services to improve the efficiency of the annotation process. Specifically, the tool uses the BSD-2 licensed core of VGG Image Annotator (VIA), which serves both academic projects and commercial applications. This is based on web technologies, making it accessible on a wide range of applications. A suite of productivity-enhancing tools has been created as a VIA extension to increase the efficiency of users (inside and outside RECLAIM) at annotating images. Notably, the entire user interface (UI) adheres to the latest design patterns, facilitating an aesthetically pleasing and user-friendly experience. This design strategy ensures that the Web application works seamlessly on various platforms, ranging from desktop computers with extended screens to the compact interfaces of mobile devices, including mobile phones. The tool is publicly available on the Reclaim-box project website at the link: https://app.reclaim-box.eu/ and is free for anyone to use.

6.2 Basic Interface Design

The annotation tool has strategically integrated external services that harness the capabilities of pretrained models, particularly Meta's "Segment Anything," for the initial annotation stage. This integration significantly empowers users by automating the segmentation of waste objects, thereby allowing them to focus primarily on correction and refinement. The tool's primary evaluation and testing focus on recyclable materials pertinent to the recycling process. While its primary aim is to assist end-users with this specific dataset, the implemented features exhibit versatility applicable to a broader range of datasets.

The annotator can use the "Segment Anything" model from Meta, that is recognized for its promotable segmentation prowess, to quickly identify the boundaries of objects. Subsequently, users are granted the capability to make necessary corrections, ensuring a fine balance between automated segmentation and user intervention. Then, the material type of the object is specified using a pop-up window that is properly structured to facilitate linking with the corresponding material type.

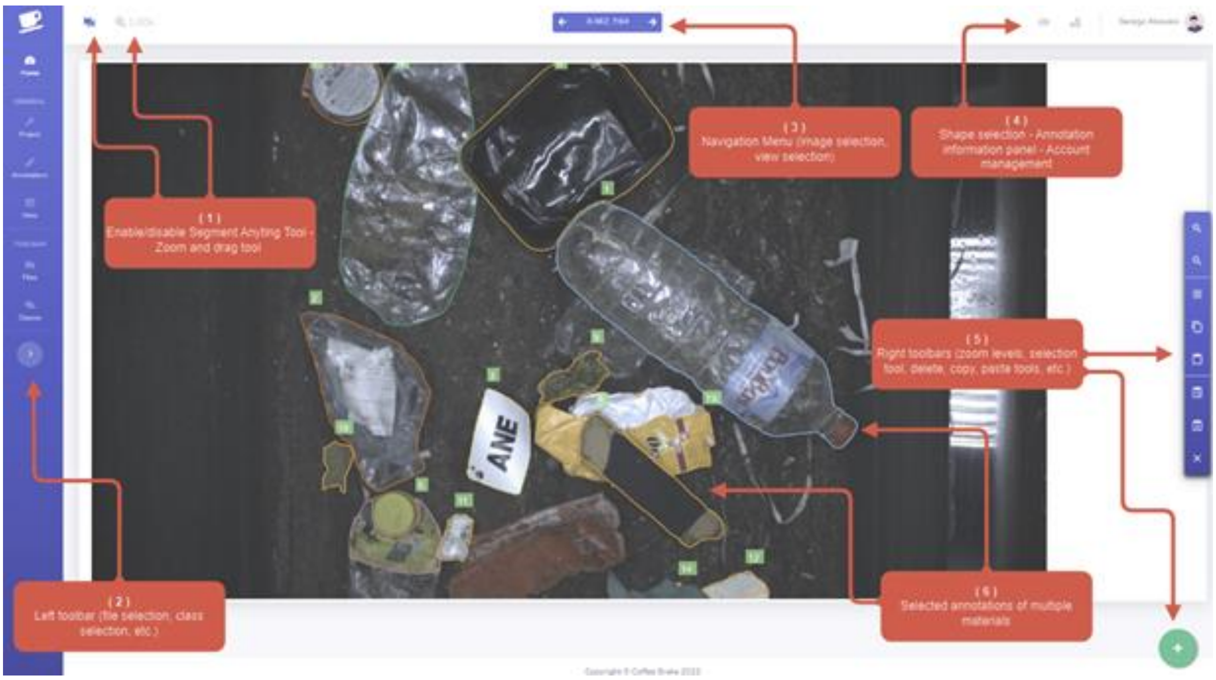


Fig. 13: Graphic interface for the annotation tool on the browser of a desktop PC.

Figure 13 shows the UI for the online annotation tool, intended for large screens and mouse controls. Below, the key UI features and functionalities are presented.

- (1) Main Tools:

- Segment Anything service enable/disable.
- Zoom control and drag control.

· (2) Left toolbar:

- Project: Load/save project and import/export file attributes.
- Annotations: Export/ Import annotation in CSV, JSON COCO format.
- View: Toggle image view to image grid, show/hide region and enable Fullscreen mode.
- Files: Import images from local storage or online.
- Classes: Add/edit/delete region attributes (classes) and image attributes.

· (3) Navigation Menu:

- Navigate to next/previous image of the list.
- Toggle image view to image grid view.

· (4) Top right menu:

- Enable/disable information of the classes.
- Shape selection tap and account manager.

· (5) Right menu:

- Annotation’s manager (delete, select, copy, paste, etc.).
- Zoom level of the current image.

· (6) Workspace: Main view of the workspace with the annotation of each material (each color represent different material).

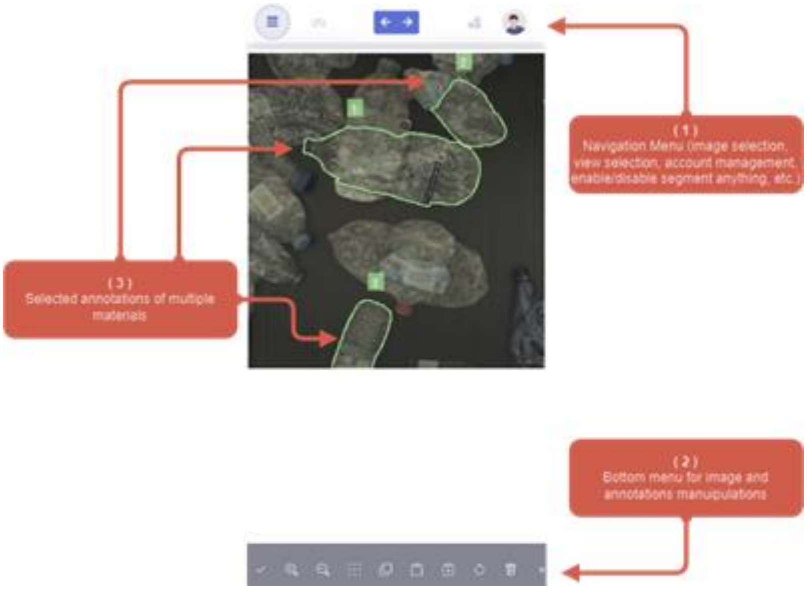


Fig. 14: Graphic interface for the annotation tool on the browser of a mobile phone.

Figure 14 shows the graphic user interface (UI) for the online annotation tool available on the phone. Below, we list the key UI features and functionalities:

- (1) Mobile top menu:
  - Segment Anything service enable/disable.
  - Zoom control and drag control.
  - Navigate to next/previous image of the list.
  - Toggle image view to image grid view.
  - Project: Load/save project and import/export file attributes.
  - Annotations: Export/ Import annotation in CSV, JSON COCO format.
  - View: Toggle image view to image grid, show/hide region and enable Fullscreen mode.
  - Files: Import images from local storage or online.
  - Classes: Add/edit/delete region attributes (classes) and image attributes.
- (2) Mobile bottom menu:
  - Annotation’s manager (delete, select, copy, paste, etc.).

  - Zoom level of the current image.

- (3) Mobile workspace: Main view of the workspace with the annotation of each material (each color represents different material).

While the RDG offers long-term motivation through gamification elements (such as questions described in Section 3 and points described in Section 4), the online tool presented here offers precision, efficiency, and adaptability. This annotation tool could be significant in preparing datasets for advanced deep learning algorithms, although it is likely to be used by a few expert users (including the RECLAIM researchers) rather than a broad population. The annotation tool is planned to incorporate more AI tools for annotation purposes and is planned to extend its functionalities to include the naming of final classes. This evolution highlights the tool's adaptability across various datasets and annotation requirements.

7. Future Work

Building off of promised future work in D6.2, we focused on the technical aspects of collecting annotations from users and the development of a central database. The technical aspects, including refinements to the actual mini-games (highlighted in Section 2.1), were crucial for evaluating the RDG under D6.3 (M18). The database allowed the collection of valuable human annotations in order to run first tests regarding (a) the user experience, potential challenges in annotating correctly, and perceived value of points (see Section 4), (b) how useful the human annotations are for AI training as new “ground truths” (see Section 4).

For the upcoming months, the results of evaluation tests (for D6.3, but also informal tests with internal RECLAIM stakeholders) will drive refinements to the technical developments described in this deliverable. The database, server, API and games will be stress-tested with many concurrent users or on poor internet connection or when deployed on older, low-spec mobile devices. Image insertion on the database on a “live” loop with the operational prMRF (see D6.4 submitted concurrently) will also test the feasibility of full-resolution image storage. As with D6.2, this will lead to refinements to all technical aspects, including bug fixes and patches.

More importantly, given our findings so far, future work will see the full implementation of the game design of RECLAIM, presented both in D6.2 and summarized in this deliverable (Section 2). We intend to revamp the interface for improving the look and feel of the game, provide a better user profile layout so that the user can track their points and feel rewarded (even if extrinsically) for their work. Importantly, we intend to include more challenges such as the content content testing challenges, which have been prepared (see Section 3) but not implemented or tested. Moreover, a level-based structure will allow for a more structured user experience, so that users are onboarded (with more explanation on material types via content testing challenges, and with simpler annotation challenges such as Detect or Count).

Finally, in tandem with development for T6.2 and evaluation tests for T6.3, we will explore ways of engaging a more general population, and focus on incentives for retaining players as they provide valuable human annotations. As per the recommendations of reviewers during the RECLAIM technical review, efforts will focus on making the RDG attractive to as broad and as diverse a population as possible. While this is challenging, given the task of annotating waste data, members of the RECLAIM consortium have a number of ideas and contacts for externally incentivizing this citizen science approach. Moreover, efforts to polish the gameplay experience and provide extrinsic motivation through user profiles, weekly challenges, and other gamification elements is expected to retain players for longer periods. The issue of intrinsic and extrinsic motivation for the RDG is expected to also be refined via human feedback during the evaluation experiments in T6.3 throughout the duration of the coming months (M18-M36). This will lead us to refine the priorities for making the RDG more attractive and guide our future development efforts.

8. Conclusion

This report presented the current state of the game design and development for the Recycling Data Game. Based on the goals of the RDG, seven mini-games for human annotation have been developed and refined in this deliverable with a structured playthrough taking the player from simpler to more complex annotation tasks (see Section 2.1). Efforts since the last deliverable (D6.2) on this task (T6.2) have focused on the infrastructure that allows the human annotations to be stored, to form a ground truth useful for AI algorithms, and to return rewards to the player in order to incentivize continued interaction and make the game more attractive in a longer term. In addition, content for awareness and quizzing has been collected and refined (see Section 3) and will be integrated in the final form of the RDG, which will focus on integrating more challenges in a more structured and engaging manner (see Section 7). Technical developments listed in this report have facilitated a functional and online game that is ready to be tested during the evaluation tasks of T6.3 and described in D6.3. Moreover, a companion app (see Section 6) will allow for more high-quality annotations by dedicated experts and citizen scientists, including players recruited via the RDG but who desire more in-depth involvement without incentives.

8. References

[Tanenbaum2013] Tanenbaum, T. J., Antle, A. N., & Robinson, J. (2013). Three perspectives on behavior change for serious games. Proceedings of the SIGCHI Conference on Human Factors in Computing Systems, 3389–3392. https://doi.org/10.1145/2470654.2466464

[Zeglen2018] Zeglen, E., & Rosendale, J. (2018). Increasing online information retention: analyzing the effects. Journal of Open, Flexible, and Distance Learning, 22(1), 22-33.

[Roediger2011] Roediger, H. L., & Butler, A. C. (2011). The critical role of retrieval practice in long-term retention. Trends in cognitive sciences, 15(1), 20-27.

[Arnab2019] Arnab, S., Lewis, M., Bogliolo, A., Klopfenstein, L. C., Delpriori, S., & Clarke, S. (2019). Player Interaction with Procedurally Generated Game Play from Crowd-Sourced data. *Extended Abstracts of the Annual Symposium on Computer-Human Interaction in Play Companion Extended Abstracts*, 333–339. https://doi.org/10.1145/3341215.3356257

[Sumner2020] Sumner, J. L., Farris, E. M., & Holman, M. R. (2020). Crowdsourcing Reliable Local Data. *Political Analysis*, 28(2), 244–262. https://doi.org/10.1017/pan.2019.32

[Strobl2019] Strobl, B., Etter, S., Meerveld, I. van, & Seibert, J. (2019). The CrowdWater game: A playful way to improve the accuracy of crowdsourced water level class data. *PLOS ONE*, 14(9), e0222579. https://doi.org/10.1371/journal.pone.0222579

~~[Foody2014] Foody, G. M. (2014). Rating crowdsourced annotations: Evaluating contributions of variable quality and completeness. International Journal of Digital Earth, 7(8), 650–670. https://doi.org/10.1080/17538947.2013.839008~~
[See2016] See, L., Mooney, P., Foody, G., Bastin, L., Comber, A., Estima, J., Fritz, S., Kerle, N., Jiang, B., Laakso, M., Liu, H.-Y., Milčinski, G., Nikšič, M., Painho, M., Pődör, A., Olteanu-Raimond, A.-M., & Rutzinger, M. (2016). Crowdsourcing, Citizen Science or Volunteered Geographic Information? The Current State of Crowdsourced Geographic Information. ISPRS International Journal of Geo-Information, 5(5), Article 5. https://doi.org/10.3390/ijgi5050055