Green Verges Project Documentation Summary

Oakleigh Weekes, Kyle Fogarty

Supervisor: Wenting Duan

Abstract

The allocation of resources to combat litter is currently a manual process. However, dashboard-cameras are being more common in motoring, and potentially offer a mechanism by which litter can be more thoroughly monitored. The Green Verge project seeks to implement a robust system that can automatically detect litter from dashcam footage, register the detected litter and geospatial coordinates, and produce a user-friendly mapping solution that highlights regions of high litter concentration. A YoloV5s model was trained on over 16k augmented images generated from over 8k dashcam frames. A baseline model was achieved which can detect litter, although further development is needed to improve upon the 0.44 mAP over a .5 IoU threshold. However, accuracy is not necessarily the objective. Plotting routes on folium maps was possible through obtaining location data via optical character recognition of dashcam footage, with areas of dense litter being linked to locations. It is recommended that a more varied data is collected and annotated, and that dashcams used for future product have location data that is directly accessible from the system itself.

Links

GitHub: Green-Verges-University-of-Lincoln (github.com)

Demo Videos: https://drive.google.com/drive/folders/19dH3SrSUh1ifSUdSCg1pyJGx_ldFa0Ae?usp=sharing

Existing Work/ Literature Review

- Litter Detection with Deep Learning: Comparative Study (Córdova M et al 2022) [Faster RCNN, Mask-RCNN, EfficientDet, RetinaNet and YOLO-v5]

- Litter Detection via UAV (Kraft et al 2021)

- Marine Litter Detection [Cascade-RCNN] (Moshtaghi, Knaeps, 2021)

- Drones for litter mapping (Andriolo et al 2021)

- Dashcam based wildlife detection (Ferreira et al 2020)

- Localize and Classify Wastes on the Streets via Street Sweeper (Liu et al 2017)

- Street Litter Detection and Classification via Edge Computing (Ping et al 2020)

- AI-based image recognition to track litter in waterways [From Hobart, to London, to Dhaka: using cameras and AI to build an automatic litter detection system (theconversation.com)]

Pipeline

- Collection of dashcam footage.
- Model Creation CI/CD:
- Manual Annotation of litter contained within the video footage.
- Training of a deep learning model to detect litter within the scene.
- Evaluation of model against unseen test footage.
- Plotting route on map.
- Plotting dense regions of litter on route.

Dataset

- Dataset consisted of dashcam footage filmed on routes around Lincolnshire.
- Consisted of ~32 hours of video.
- Footage was taken during February and March time, covering conditions such as dark, dull and light and rain weather.
- 4K footage was downgraded such that the chosen annotation software (CVAT) can handle parsing through the frames.

Figure 1 Varied dataset samples (weather / light conditions/ location)

Annotation

- It is necessary to note annotation could introduce error- whether zooming in is necessary to fit litter boundary exactly. A strict annotation policy needs to be established.
- CVAT – Open source annotation tool developed by Intel [openvino/cvat_ui - Docker Image | Docker Hub](#) .
- Contains useful options such as bounding box interpolation through frames (i.e. shifts bounding box around litter according to its distance moved along the road as the dashcam's vehicle passes it).
- In total, 8997 frames were annotated.

Preprocessing

Removing Redundancy

- When annotating raw dashcam video, there may be many frames that do not contain annotations.
- In the case of the object detector, these empty frames are redundant as the model will not learn much information from them with regards to classes and position.

- CVAT outputs two things:
  - All the frames in the imported video (Including those without litter)
  - Json annotations file (Includes references to empty frames).
- CVATHelpers was created to filter frames and annotations. [Green-Verges-University-of-Lincoln/CVATHelpers: CVAT COCO exported dataset manipulation (github.com)](https://github.com/Green-Verges-University-of-Lincoln/CVATHelpers)

- CVATHelpers has separate functions for filtering json and frames, and 'reduceCVAT' which accomplishes both at the same time.
- However, the solution only works on annotations in COCO format.

Augmentation

- Roboflow (https://app.roboflow.com/) free tier was used to:
  - Convert annotations to YOLOv5 format
  - Resize images 640 x 640 (Optimum recommendation for Yolo)
  - Double the dataset with Blurred Images. This was in an attempt to simulate motion blur which will affect dashcam frames as the vehicle collects data.
- The total training dataset thereby rose to 16,194 images.
- Roboflow free tier supports 10,000 uploaded images and a certain amount of transformed augmentations.

Deep Learning Model

YoloV5

- YOLO ('you only look once') is an object detection algorithm that divides into a grid system, with each cell responsible for detecting objects within itself. It is known for its speed and accuracy.

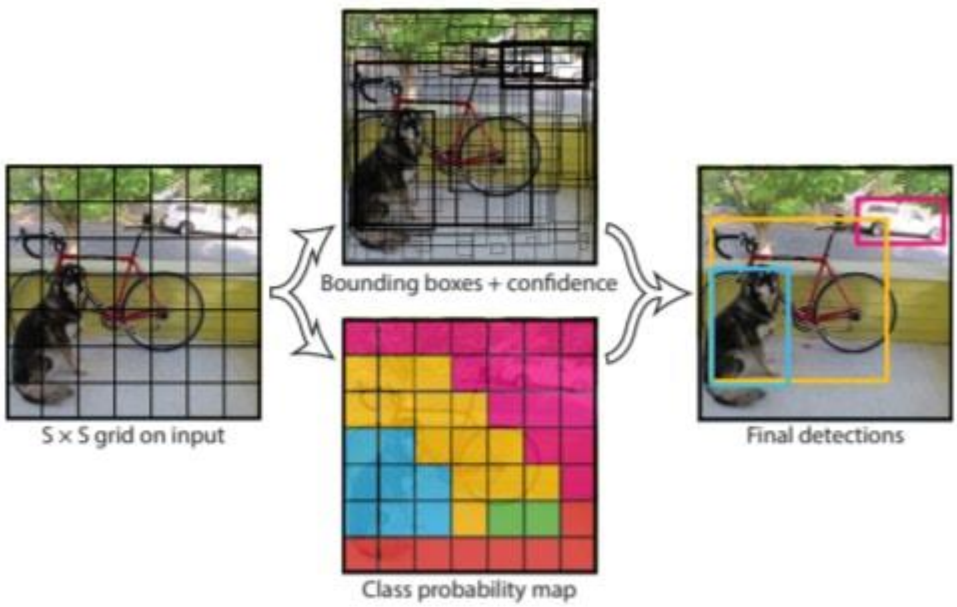

Figure 2 Summary of the YOLO algorithm

[Introduction to YOLO Algorithm for Object Detection | Engineering Education (EngEd) Program | Section]

- V5(s) was introduced by Glenn Jocher and uses the Pytorch network. It has an active development community which proved useful in debugging. Some problems (such as installation compatibility issues) were ironed out by browsing the github forums (which are very ongoing and seemed to almost answer any questions in real-time; it being still a new algorithm released initially in May 2020). YOLOV6 has arguably been released very recently, however, has very little documentation or community surrounding it, and even less research (papers) and development. After development V7 was released and may be beneficial to use (Wang et al 2022).

Training Method

- The Weights & Biases API was used to view resource use and training metrics. The resource graphs helped to realise when the model was not being trained on the GPU (necessary), or the batch size was too big for the GPU and memory to handle (Batch size of 2 fell optimum in the 3301 Office PC GPUs). The training visualisations informed when there was an error in the dataset annotations, and when overfitting started to occur.
- The YOLOv5(s) framework allows for the epoch with the best accuracy values to be saved.
- The model was first pretrained with the aerial dataset (UAVVaste/UAVVaste: UAVVaste: COCO-like dataset and effective waste detection in aerial images (github.com)) .

Figure 3 UAV litter dataset with which the model was pre-trained

- Through transfer learning, the model was brought forward for further training on the augmented dashcam dataset, taking around 2 days of training.

Detection Outcomes

- The model was tested on annotated unseen video and achieved a 0.44 mean average precision over a .5 IoU threshold and mAP of 0.13 over a threshold range of [.5:.95].
- The model began to overfit after epoch 40.
- The trained model prediction output shows that litter detection is a complex problem. The model needs to recognise all road environments and all litter types, big and small. A possible remedy would be to collect a very large, varied dataset. This is further stressed if litter is split into classes such as paper, plastic etc.
- It needs to be discussed whether increasing accuracy substantially would be worth the extra resources required. The objective for this project is to locate dense areas of litter, and so having a predicted bounding box that does not fit a litter shape perfectly is not of importance. However, removing the existence of false positives is required, and accuracy improvements need to address this.

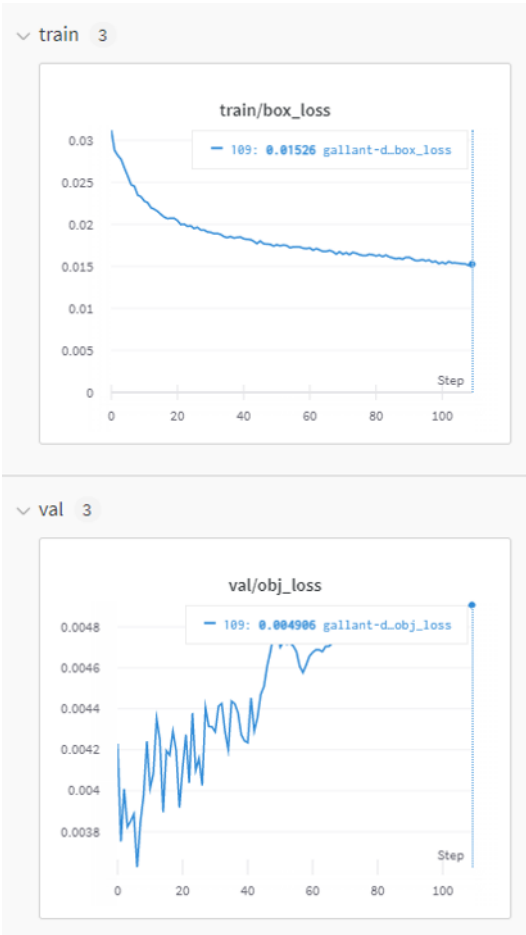

Figure 4 Training loss flattening and Validation loss Increasing after epoch 40 (Overfitting)

Figure 5 Model correctly detecting litter

Figure 6 Detection model detecting litter and falsely identifying a sign as such

GPS extraction & Route Logging

- Ideally, GPS data would be extracted via the webcam.
- Our dashcam camera was too heavily encoded, making it difficult to obtain clear data.
- However, GPS data (Latitude, Longitude & Speed) can be seen embedded visually in video (and its frames).
- OCR (Optical Character Recognition) was used to extract this data.
- A nanonet was trained originally and used until it was found that after a certain amount of API calls it expends the free tier's resources.

- Instead, Tesseract - in particular ‘pyTesseract’ - was used as a pre-trained model to detect these 3 variables (Latitude, Longitude & Speed) so they could be used for plotting route and litter location.
- A subsample of frames was selected from a video (one 2-minute video may contain over a thousand).
- The process involved cutting the variables out of frame, image processing and filtering for correct prediction through string formatting.

- Processing consisted of grayscale conversion, image inversion, contrast enhancement, and binarization.
- After prediction, string formatting made sure that the long/lat format was obeyed, such as correct direction letter (N/S/E/W), point (.) and fixed issues such as mistaking ‘8’ for ‘B’ and ‘O’ for ‘0’.
- Duplicate long/lat combinations were also removed.
- The Folium library was used to visualise routes and maps using python.
- Paths were created through linking subsequent GPS detections, e.g.
  `[(53.449, -1.11), (53.448, -1.109), (53.448, -1.108), (53.446, -1.107), (53.442, -1.103), (53.441, -1.102), (53.44, -1.101), (53.413, -1.083), (53.411, -1.082)]`

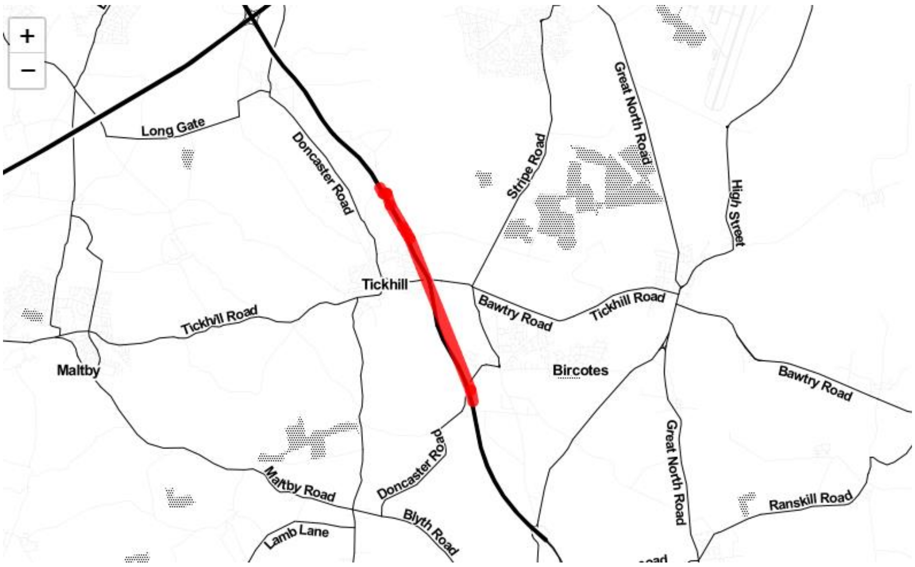

Figure 7 Mapped route using obtained locations and Folium

Litter Flagging

- The final stage was to fuse the route logging with the object detection model. The video footage of interest was passed through both the OCR pipeline and the object detection pipeline.
- The object detection model returns bounding box size (which we convert to a normalised area) and the frame that bounding box was in; this information is parsed from the object detection output by a python script and saved to disk.
- The route logging only provides a finite number of waypoints – to plot the approximate location of litter we perform a linear interpolation between waypoints based on frame IDs. (If frame 0 and 10 are waypoints and we need the location of

frame 7 we place that location at 70% of the distance between the line segment connecting frame 0 and 10).

- Once approximate latitude and longitude location have been found for each object detected, they are published to the Folium map as a heatmap dependent upon the bounding box area.
- Frames from the object detector (I.e., with bounding boxes drawn) are saved to disk and linked to the Folium map as a popup for each region such that litter can be inspected.

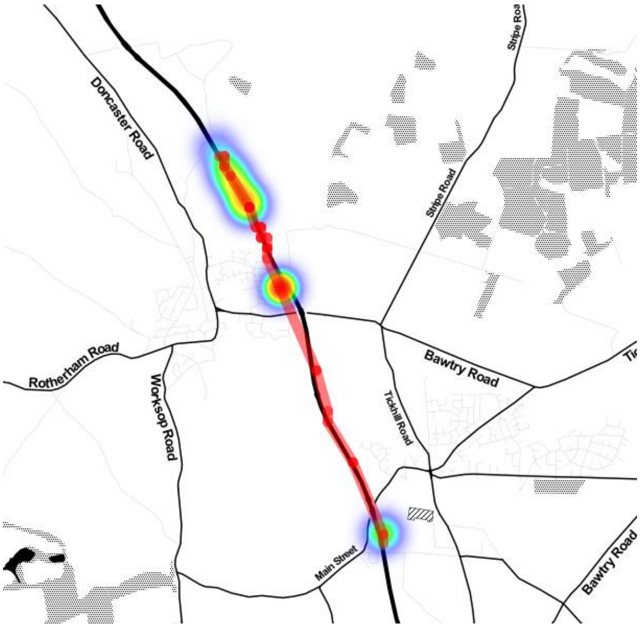

Figure 8 Litter heatmap example extracted from a detection pipeline.

Evaluating other found requirements

Dataset

- Acquire a larger, varied dashcam footage dataset.
- Create a program which resizes data, converts it to Yolo format and performs augmentation (This will be needed as RoboFlow's free tier acts as a limitation to large datasets).

Computational

- Obtain upgraded computational resources to train deep learning models more efficiently.
- Use evolution techniques to find optimum YoloV5(s) parameters (requires 'beefy' hardware).
- Compare and contrast various models such as R-CNNs and Transformers.

Annotation Workflow

- Class litter into sub-categories and train a new model to learn these (requires specialised annotation).
- Label litter features- based on size and material

Litter Logging

- State which side of the road the litter is located. Dashcams focus more so on the left side of the road (road law), as the right side is further away/ obscured by vehicles. To note which side of the road the litter was located:
  - From a detection standpoint- include which side of the frame the litter is located on (left, right, mid).
  - From a route logging standpoint, log the direction the vehicle is driving (could be inferred by the change in longitude and latitude values).

Branched Projects

- Log Potholes (based on keep Britain Tidy's advice)

Conclusion

A conceptual solution has been created which detects litter on roads with a 0.44 mAP accuracy over a .5 intersection-over-union threshold and logs this detection on a heat mapped route based on litter sizes. This was made through transfer-training of a YoloV5s network and harnessing location variables through optical character recognition. Although a balance needs to be had with increasing model accuracy, further training on an increased varied dataset could reduce the number of false positive detections. The project can also be extended in numerous ways, such as returning more information about litter type and location to the user.

References

Córdova M., Pinto A., Hellevik CC., Alaliyat SA-A., Hameed IA, Pedrini H. and Torres RdS. (2022) Litter Detection with Deep Learning: A Comparative Study. *Sensors*. 2022; 22(2), 548. Sensors | Free Full-Text | Litter Detection with Deep Learning: A Comparative Study | HTML (mdpi.com)

CVATv2.1.0 openvino/cvat_ui - Docker Image | Docker Hub

Kraft, M., Piechocki, M., Ptak, B. and Walas, K. (2021) Autonomous, Onboard Vision-Based Trash and Litter Detection in Low Altitude Aerial Images Collected by an Unmanned Aerial Vehicle. Remote Sens, 13, 965.

Liu M., Chen H. and Vincze M. (2017) (eds) Computer Vision Systems. pp 195-204. ICVS 2017. Lecture Notes in Computer Science, vol 10528. Springer, Cham

Ping, P., Xu, G., Kumala, E. and Gao, J. (2020) Smart street litter detection and classification based on Faster R-CNN and edge computing. Int. J. Softw. Eng. Knowl. Eng. 30(04), 537–553

Wang CY, Bochkovskiy A and Liao HYM (2022) YOLOv7: Trainable bag-of-freebies sets new state-of-the-art for real-time object detectors [2207.02696] YOLOv7: Trainable bag-of-freebies sets new state-of-the-art for real-time object detectors (arxiv.org)

Zhu X, Lyu S, Wang X and Zhao Qi (2021) TPH-YOLOv5: Improved YOLOv5 Based on Transformer Prediction Head for Object Detection on Drone-captured Scenarios 2108.11539.pdf (arxiv.org)

Ferreira BA., de Villiers JP. and de Freitas A. (2020) Dashcam based wildlife detection and classification using fused data sets of digital photographic and simulated imagery. *2020 IEEE 23rd International Conference on Information Fusion (FUSION)*. 1-8. Dashcam based wildlife detection and classification using fused data sets of digital photographic and simulated imagery | IEEE Conference Publication | IEEE Xplore (lincoln.ac.uk)

Andriolo, U. et al. (2021) Drones for litter mapping: An inter-operator concordance test in marking beached items on aerial images, *Marine Pollution Bulletin*, 169.

Drones for litter mapping: An inter-operator concordance test in marking beached items on aerial images (sciencedirectassets.com)

The Green Verge Project

Litter awareness from GPS enabled dashcam footage

Oakleigh Weekes, Kyle Fogarty, Dr Wenting Duan†

Laboratory of Vision Engineering, School of Computer Science, University of Lincoln

Abstract

The allocation of resources to combat litter is currently a manual process. However, dashboard-cameras are being more common in motoring, and potentially offer a mechanism by which litter can be more thoroughly monitored. The Green Verge project seeks to implement a robust system that can automatically detect litter from dashcam footage, register the detected litter and geospatial coordinates, and produce a user-friendly mapping solution that highlights regions of high litter concentration.

Introduction

- Littering has **detrimental effects** on both the **natural environment**, with the breakdown of litter releasing chemicals, and the wildlife that inhabits it [2].
- **Litter** has been shown to be an **increasing problem** in the **U.K** [3].
- The cost of combating **litter costs** an estimated **£700 million a year** to local authorities [1].

Recent consultations with **local authorities** highlighted a **demand for** the development of a **computer vision** based system that can process, and **identify litter** within video footage to allow for **better allocation of resources**.

- **Dashboard-cameras** (dashcams) have become fairly **ubiquitous** in modern motoring, being fitted as standard in many new cars. Furthermore, they are **already fitted** to many **local authority vehicles**, like refuse collection trucks.
- Dashcams provide **inexpensive** mechanism for **autonomously collecting video footage** from the environment, and many are **GPS enabled**.

In this project, we seek to build a ***deep-learning*** based system that is able to accept GPS embedded vehicle dashcam footage, and **robustly identify litter** within it.

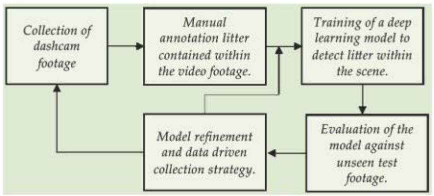

Figure 1. Overview of training pipeline adopted in the green verge project.

Work Overview

The Green Verge project has been split into work packages (WPs); in this poster we present the work conducted to-date on WP1: ***dataset creation and model training***. The training pipeline adopted for WP1 as show in figure 1. Our main aims in this work are:

1. **Construct** an annotated **dataset of litter** from dashcam footage.
2. Use our dataset to **training** a **deep learning model** to detect litter in dashcam footage.

Future work will consider WP2: *litter visualisation and mapping*.

Dataset Construction

Our dataset is constructed from 4K dashcam footage, collected in Lincolnshire between February and April 2022. Video footage was annotated with bounding boxes, for the single class we refer to as `litter`, using the open-source annotation tool CVAT [6]. We also label each video according to the weather and lighting conditions seen, to aid with our data driven collection strategy (Fig. 1).

In **total**, our dataset consists of **8997 frames** (constructed as a subset of the **32 hours** of total footage captured). The size of our dataset is compared to related works in figure 2. In our work, we also consider **augmentation techniques**, like **blurring**; in doing so, we increase the size of the training dataset and include domain specific inductive bias (e.g. vehicle motion blur), without the costly need of annotating further frames. When including **augmented** frames, our dataset contains **15,065 frames**.

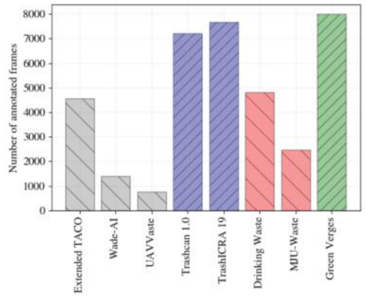

Figure 2. Comparison of the size of our dataset compared with related works; grey columns represent datasets with litter captured in urban/outdoor environments, blue represents datasets captured underwater, and red represents datasets captured indoors.

Deep Learning Model

Our recent work has been to obtain a **baseline implementation**, before we begin experimenting with different architectures. To establish our baseline we considered the **YOLO framework** [5]. We used weights pretrained from the COCO dataset, with fine-tuning conducted on the UAVVASTE dataset [4]. After training the YOLO network on our dataset for 109 epochs, **our network achieved mean average precision (mAP), at a 50% intersection over union threshold, of 44%.**

The main challenges we will seek to address in future work is: 1) improving the mAP achieve such that it is deployable in practice, 2) making the network more robust to detecting smaller object, and 3) improving robustness to different lighting conditions, reflections, and dynamic objects (like cars and pedestrians) in the scene.

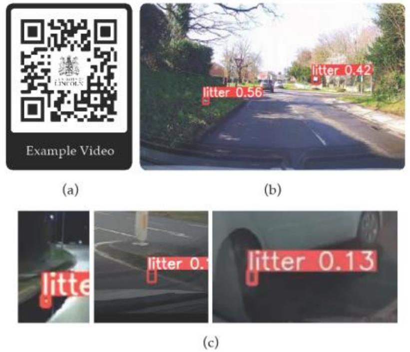

Figure 3. (a) Example test video with our current implementation. (b) Example of where litter is correctly and incorrectly detected in a single frame. (c) Examples of challenging frames for the network to deal with: reflections in water, road markings, moving vehicles.

References

[1] Department for Environment, Food & Rural Affairs. Litter and littering in England 2018 to 2019. https://www.gov.uk/government/publications/litter-and-littering-in-england-2017-to-2018. Online; accessed 1 June 2022.

[2] Frederic Gallo, Cristina Fossi, Roland Weber, David Santillo, Joao Sousa, Imogen Ingram, Angel Nadal, and Dolores Romano. Marine litter plastics and microplastics and their toxic chemicals components: the need for urgent preventive measures. *Environmental Sciences Europe*, 30(1):1–14, 2018.

[3] Keep Britain Tidy. The local environment quality survey of England 2017/18. https://www.keepbritaintidy.org/news/survey-reveals-litter-increase, note = Online; accessed 1 June 2022.

[4] Marek Kraft, Mateusz Piechocki, Bartosz Ptak, and Krzysztof Walas. Autonomous, onboard vision-based trash and litter detection in low altitude aerial images collected by an unmanned aerial vehicle. *Remote Sensing*, 13(5), 2021.

[5] Joseph Redmon, Santosh Divvala, Ross Girshick, and Ali Farhadi. You only look once: Unified, real-time object detection, 2016.

[6] Boris Sekachev, Nikita Manovich, and Andrey Zhavoronkov. Computer vision annotation tool, October 2019. GitHub: https://github.com/opencv/cvat.

†Corresponding author: wduan@lincoln.ac.uk

Keep Britain Tidy

- A project presentation was given by Wenting Duan, Kyle Fogarty and Oakleigh Weekes to the Keep Britain Tidy committee.
- The presentation covered previous detection endeavours, data collection, and how artificial intelligence can aid councils in having an objective litter flagging solution.
- At the time of presentation, an accurate model was not yet produced. However, the team has now kept contact for KBT to hear of the finished research demonstration.