# Automated trash screen blockage segmentation using deep learning

Remy Vandaele[1]
r.vandaele@exeter.ac.uk
Sarah L. Dance[2,3,4]
s.l.dance@reading.ac.uk
Hywel T.P. Williams[1]
h.t.p.williams@exeter.ac.uk
Varun Ojha[5]
varun.ojha@newcastle.ac.uk

[1] Joint Centre of Excellence in
Environmental Intelligence
University of Exeter
Exeter, UK

[2] Department of Meteorology
University of Reading
Reading, UK

[3] Department of Mathematics and
Statistics
University of Reading
Reading, UK

[4] National Centre for Earth Observation
University of Reading
Reading, UK

[5] School of Computing
Newcastle University
Newcastle-Upon-Tyne, UK

**Abstract**

Trash screens are used to prevent floating debris from damaging critical assets (e.g. pipes, pumping stations) in rivers. However, debris accumulates at the trash screen location and can contribute to floods. Here we develop a novel application of deep learning that uses cameras to automatically monitor the presence and amount of trash on trash screens. We manually annotated debris in 575 trash screen images from 54 cameras and used this dataset to train and evaluate the performance of several semantic segmentation networks. This process reaches segmentation accuracy above 95% MIoU using the SegVit network based on a Vision Transformer architecture. We show that this approach can be used to accurately monitor the state of trash screens during flood events, detecting build up of trash to guide preventative maintenance. This research is an important step towards the automation of trash screen monitoring, an application of great importance in environmental monitoring and better management of flooding.

## 1 Introduction

The presence of debris (trash) in rivers is a critical problem in river management. If left unattended, debris can cause damage to river assets (pipes, pumping stations) and cause floods [12]. Thus debris is typically stopped at strategic river locations by trash screens [2].

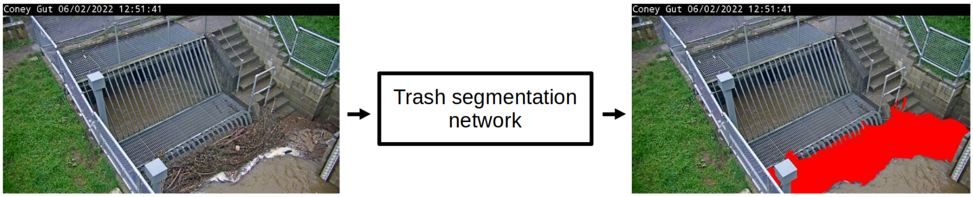
Coney Gut 06/02/2022 12:51:41
Trash segmentation network
Coney Gut 06/02/2022 12:51:41

Figure 1: Representation of the trash segmentation process. The trash segmentation takes as input an image (in this case, a trash screen image with debris accumulating at the lower screen) and outputs a mask of the debris (in red).

As shown in Figure 1, trash screens are assets made of vertical bars with spacing designed to prevent the relevant debris from passing through. However, once debris starts gathering at the trash screen, the blocked debris itself will start to block additional debris, which will thus accumulate at the trash screen location until the trash screen eventually gets obstructed. This prevents the water from flowing normally and can contribute to floods or nullify the purpose of the trash screen by allowing the debris to flow above the screen itself. This is why it is extremely important to clean the trash screens.

Currently, scheduling the cleaning of such screens is difficult as they may not be remotely monitored. This leads to unnecessary manual inspections and labour costs, while the related loss of time can become critical in emergency situations.

A new approach to monitor the state of these screens has been to use cameras to look at the screens. Cameras are relatively cheap, and can transmit images of the screens through cellular broadband connections. In the UK, the South West Environment Agency has installed around 350 cameras to monitor trash screens installed on rivers [5]. However, monitoring the cameras manually can still be complex with high numbers of trash screen cameras to monitor.

This study focuses on the development of a novel application of deep learning to monitor the state of these trash screens using camera images. As shown in Figure 1, by detecting the location of the debris on the screen, it becomes possible to automatically and accurately monitor the state of the trash screens. The approach is based on the training of a deep semantic segmentation network on a dataset of trash screen camera images.

The key contributions of this study are:

1. A dataset of 575 trash screen camera images labelled with pixel-wise segmentation of debris is made available online [1].

2. The creation and evaluation of two deep learning models able to segment (and quantify) the amount of trash on the screens.

3. The development and validation of an automatic system leveraging the deep learning models to accurately monitor the state of the trash screens, reducing the time for manual inspection.

## 2 Related Work

Attempts to predict trash screen blockage exist in the literature. Wallerstein et al. [17] conducted an experiment during which they manually observed 140 trash screens around

Belfast (Northern Ireland) between 2002 and 2008. They concluded that certain catchment configurations were more prone to blockage, with rainfall and the time of year being the main drivers. In a follow-up study, Wallerstein et al. [16] proposed monthly equations to predict the probability of blockage, considering parameters such as channel slope and rainfall. The accuracy of these equations was later improved using a Bayesian model [13]. However, the authors noted that the lack of data prevented them from fully verifying their models and that their dataset included only urban trash screen observations, biasing the experiments towards urban areas. To the best of our knowledge, there is no evidence that these works have been used in practice to aid trash screen maintenance planning.

The use of deep learning to detect and classify trash on camera images is being extensively studied. For example, Jaikumar et al. [8] used a Mask R-CNN to detect water bottles in camera images. Tharani et al. [14] and Nguyen et al. [11] used object detection networks to detect floating trash on water surfaces. Majchrowska et al. [9] offer a comprehensive study of the most common datasets used in this field. These works differ from our task in two main aspects. Firstly, they consider anthropogenic waste (e.g. plastics, metals) and ignore other types of trash that commonly block screens (e.g. tree branches). Secondly, these studies focus on placing individual bounding boxes around the trash objects. In our case, such rectangular bounding boxes would provide an inaccurate quantification of the trash blocking a screen, as it would be highly dependent on the field of view of the camera as well as the position and orientation of the trash on the screen.

The detection of trash screen blockage using cameras is thus a new challenge. Iqbai et al. [6] compared the efficiency of image classification with several convolutional neural networks (CNNs) to separate *blocked* culvert images from *clear* ones. The authors noted that the synthetic nature of most of their images could impact the results and prevent generalisation. In a subsequent study, Iqbal et al. [7] attempted to estimate the percentage of hydraulic blockage from culvert images. However, the authors were limited to images captured in a hydrology lab in the same controlled environment as in their previous study.

More recently, Vandaele et al. [15] tested different approaches to classify *blocked* and *clean* trash screen images coming from real trash screen camera data. While they obtained very promising results, they noticed that the binary classification of such images was confused by the subjective labelling of borderline cases where it was unclear whether the trash was actually blocking the screen. They also collected, labelled and shared a dataset made of 80,452 images coming from 54 different trash screen cameras.

Following the analysis of the related work, we decided to tackle the trash screen monitoring problem with semantic segmentation networks that would detect the amount of debris located in the image, as depicted in Figure 1. In the context of trash screen monitoring, a pixel-wise segmentation of the debris will bring information that is less subjective than a binary classification assigning a *blocked* or *clean* label assigned to the whole image. Also, this approach will provide information that is more accurate than an object detection approach delineating rectangular bounding boxes around the debris.

# 3 Methodology

This section outlines the methodology used to build the trash semantic segmentation networks, the creation of a labelled dataset, and the choice of semantic segmentation models.

| | Clear | Other | Blocked | Total |
|---|---|---|---|---|
| Training cameras | 96 | 0 | 279 | 375 |
| Test cameras | 91 | 46 | 63 | 200 |

Table 1: Number of images manually annotated with pixel-wise trash labels. Images and blocked/clear/other labels taken from Vandaele et al. [15].

## 3.1 Dataset

We used the images from the dataset created in Vandaele et al. [15]. This dataset is based on 54 cameras and contains 80,452 daylight RGB images labelled *blocked* when debris was blocking the screen, *clear* if there was no debris or the debris was not blocking the screen, and *other* if the human annotators were not sure. To facilitate the classification task, images were then cropped to the region of the trash screen. The authors supply both the original full size images and the cropped locations. Among the cameras, 4 are used for testing, and the remaining 50 for training and validation. Here we intended to compare our segmentation approach with the classification approach presented by Vandaele et al. [15], so we have kept the same cameras for training and testing.

As the pixel-wise manual labelling of images is a time-consuming task, we chose to reduce the number of training and test images used to train and test our segmentation networks. However, note that some of the comparisons with Vandaele et al. [15] that are presented in Section 4 rely only on the image labels and not the pixel-wise segmentation. In these cases, the entire test set was used; these cases are identified in the results below.

We initially annotated 5 randomly selected images labelled as *blocked* for each of the 50 training cameras. This gave an initial training dataset of $5 \times 50 = 250$ training images. As the regions processed by the networks are cropped from the original images, instead of generating a single crop like Vandaele et al. [15], here we generated 10 crops per image using small random variations with Vandaele et al's crop sizes (from 90% to 110% of the original crop height/width) and locations ($x$ and $y$ coordinates of the upper left corner moved within a range of $-10\%$ to $10\%$ of the full size image width/height). This created an initial dataset of $250 \times 10 = 2500$ crops.

With this initial dataset, we trained the ResNet50-UperNet trash segmentation network using the protocol described in Section 4.1. We then applied the trained network to a dataset composed of 50 images per training camera (from Vandaele et al's [15] crops) that were randomly selected independently of their blocked/clear label. Any images on which the trained network made obvious segmentation mistakes were then added to the segmentation training dataset and manually annotated pixel-wise to identify trash. These images were then added to the training set by using the data augmentation process described in the previous paragraph (10 crops were generated for each new image).

For the 4 test cameras, we labelled 50 images per camera, independently of their classification label, using the same crops as Vandaele et al.[15]. This gave $50 \times 4 = 200$ test images and crops.

In total, the process above created a dataset of 375 training images (augmented to 3750 cropped images) and 200 test images (which were not augmented). More details are given in Table 1. This dataset is openly accessible online [1].

## 3.2 Segmentation networks

We tested two semantic segmentation networks: one based on a traditional convolutional neural network, and one based on a more recent vision transformer (ViT) architecture.

For the convolutional neural network, we chose to adopt a ResNet50-UperNet (UperNet50) architecture [20]. ResNet-50-UperNet is based on an encoder-decoder architecture. The encoding is done through a ResNet-50 network. The decoder is UperNet [18], which is based on a Feature Pyramid Network (FPN) and Pyramid Pooling Module (PPM) to integrate multi-scale contextual information obtained at different layers of the encoder network. This architecture obtains among the best results on the ADE20k semantic segmentation dataset [20]. We used the CSAILVision implementation of this network [20].

For the vision transformer architecture, we used SegViT developed by Zhang et al. [19]. SegViT is also based on an encoder-decoder architecture. The encoder is a plain Vision Transformer (ViT) model that splits the image into patches which are further transformed into tokens (one-dimensional vectors) through multiple transformer layers (see Dosovistkiy et al. [4] for more details). The decoder is based on the sequential application of the Attention-To-Mask (ATM) module to each token produced by the encoder [19]. The ATM module produces one mask per token using a self-attention mechanism, which is then combined to produce the final mask. Zhang et al. were able to show that their network outperformed the state-of-the-art networks on three well-known segmentation datasets (COCO-stuff [3], ADE20k [20] and Pascal-Context [10]). We used the SegViT implementation code proposed by the authors on GitHub [19].

# 4 Results and experiments

## 4.1 Segmentation results

The goal of this experiment was to train and evaluate the semantic segmentation networks presented in Section 3.2 with the dataset that we created and described in Section 3.1.

**Experimental setting.** We used a training/validation and test protocol similar to Vandaele et al. [15] who used the same camera dataset: we used the same 4 test cameras (Barnstaple, Crinnis, Mevagissey and Siston) for testing, and the remaining 50 cameras for training and validation. We used one randomly selected image per training camera for validation (50 images) and used the remaining images for training. We trained each network using the training parameters recommended by [20] and [19] respectively, except for the learning rate for which we tested values $10^{-3}, 10^{-4}$ and $10^{-5}$. The networks were trained over 30,000 iterations (steps), with two images processed per step. Each 1000 steps, the network was validated on the validation set. The network weights at which the validation accuracy was higher were kept. Note that both implementations use data augmentation (random cropping and flipping).

**Evaluation Metric.** The Mean Intersection over Union (MIoU) was chosen as the primary metric for evaluating the segmentation performance of our models. The IoU for each class (debris, or no debris in our case) is calculated by dividing the intersection of the predicted and ground truth areas by their union. This approach provides a clear and balanced

| | ResNet50-UperNet [20] | SegViT [19] |
|---|---|---|
| Barnstaple | 74.2 | 77.5 |
| Crinnis | 80.8 | 84.6 |
| Mevagissey | 52.2 | 68.2 |
| Siston | 66.4 | 80.8 |
| *Average* | 68.4 | 77.8 |

Table 2: MIoU segmentation results for the two networks on four test camera locations.

measure of both false positives and false negatives, ensuring that the metric accounts for over-segmentation and under-segmentation equally.

Pixels classified as debris are labelled as class 1 and pixels with no debris are labelled as class 2. Then the IoU for each class $i$ is defined as:

$$\mathrm{IoU}_i = \frac{TP_i}{TP_i + FP_i + FN_i} \tag{1}$$

where $TP_i$ is the number of true positives for class $i$, $FP_i$ is the number of false positives, and $FN_i$ is the number of false negatives. The MIoU is then computed as the average of the IoUs for all classes:

$$\mathrm{MIoU} = \frac{1}{2}\sum_{i=1}^{2} \mathrm{IoU}_i \tag{2}$$

**Results.** The MIoU segmentation results are given in Table 2. On average, SegViT obtains better MIoU scores than ResNet50-UperNet. However, while SegViT obtains better results than ResNet50-UperNet at every location, the difference seems to be particularly important at Mevagissey and Siston. Mevagissey is the hardest location for both networks, while Crinnis is the easiest. We give some examples that we deem representative of the results in Figure 2. At Barnstaple, both networks tend to have difficulties with the detection of the smaller debris. At Crinnis, both networks segment the debris correctly. At Mevagissey, ResNet50-UperNet tends to miss white plastic debris common to this screen while confusing the trash screen bars for debris. At Siston, ResNet50-UperNet struggles with the detection of particular debris (e.g., tyres or plastics).

In conclusion, our main observations are that:

- Both networks tend to generally correctly segment most debris at the 4 screen locations
- Both networks tend to struggle with the detection of the smaller debris
- The main difference between SegViT and ResNet50-UperNet is that SegViT tends to work better on types of debris less represented in the training set, like plastics and tyres.

## 4.2 Debris monitoring and comparison with Vandaele et al. [15]

### 4.2.1 Classification of trash screen camera images

The goal of this experiment is to compare the performance with the classification approaches used in Vandaele et al. [15] to make the distinction between *clean* and *blocked* trash screens.

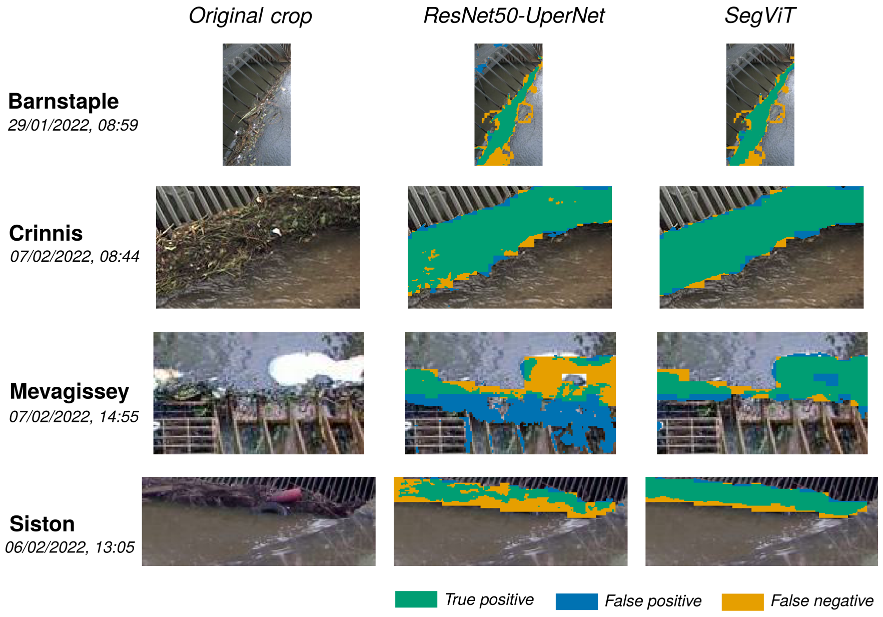

Figure 2: Representative examples of the segmentation results obtained at the four test locations.

For this new experiment, we have applied our trash semantic segmentation networks with their best performing network weights obtained in the first experiment (see Section 4.1) on the entire set of test camera images produced by Vandaele et al. [15] and not only the 50 that we labeled with segmentation annotations. Indeed, Vandaele et al. labeled these images with class annotation *clear*, *blocked* or *other* and evaluated different methods to classify the *clear* and *blocked* images. These methods all produce a *blockage score* that can be thresholded to classify the image into one of these two classes.

With our segmentation method, we can define the *blockage score* as the percentage of pixels of the image that is detected as containing trash, and compare our method with Vandaele et al.'s method [15] in terms of classification accuracy. For this comparison, we used the ROC AUC score used in Vandaele et al. [15]. In binary classification, A ROC curve plots the true positive rate (TPR) against the false positive rate (FPR) at different classification (in our case, blockage score) thresholds,

$$\mathrm{TPR} = \frac{\mathrm{TP}}{\mathrm{TP}+\mathrm{FN}},\ \mathrm{FPR} = \frac{\mathrm{FP}}{\mathrm{TP}+\mathrm{FN}}, \tag{3}$$

and the ROC AUC score computes the area under that curve. An area of 1 means that no matter the threshold, positive (blocked) and negative (clean) samples are perfectly separated. An area of 0.5 means that the classification is random.

For our comparison, we only considered Vandaele et al's best performing method, which is a Siamese network that computes the blockage difference between 5 labelled reference images of the test camera, and a new image of that camera.

The results are shown in Table 3. In average, our segmentation network based on Vision Transformers, SegViT, performs better than Vandaele et al.'s best performing method. How-

| | Vandaele et al. [15] | ResNet50-UperNet | SegViT |
|---|---|---|---|
| Barnstaple | 95.6 | 99.5 | 99.8 |
| Crinnis | 98.1 | 98 | 96.7 |
| Mevagissey | 99.1 | 92.4 | 99.1 |
| Siston | 97.2 | 94.3 | 95.4 |
| *Average* | 97.5 | 96.1 | 97.7 |

Table 3: Comparison of the performance (ROC AUC scores) of our methods with Vandaele et al.'s Siamese network for classifying clean and blocked trash screen images.

ever, it is outperformed at 2 out of the 4 locations. The ResNet50-UperNet network obtains slightly lower performance than Vandaele et al., except at the Barnstaple location. We note that our methods were not trained for the classification of blocked and clean images, and were trained on much smaller datasets (375 images instead of 80,000). However, they are competitive with the classification approaches proposed by Vandaele et al. [15].

## 4.3 Automated monitoring of a blockage event

With this experiment, we want to evaluate the potential utility of using the segmentation *blockage score* to monitor the evolution over time of trash screen blockage on the screens, and explore whether it could contribute to an efficient alarm system.

In Section 4.2.1, we define the segmentation blockage score as the percentage of a trash screen image estimated as containing trash. As we do not have ground truth segmentation data for the entire period, we consider whether this blockage score is able to capture the progressive increase of trash screen blockage over time before the screen gets fully blocked. We also compare performance against the blockage scores provided by the image classification methods proposed Vandaele et al. [15].

For each of the test cameras, we looked for a 7 to 10 day period during which we could witness a blockage event on the camera images. We then analysed the corresponding evolution of the blockage scores from our segmentation methods and the blockage score produced by the Siamese network of Vandaele et al. [15]. The results are provided in Figure 3.

At Barnstaple, the segmentation networks are able to capture the progressive evolution of the blockage on the screen, while the Binary Classifier struggles (Barnstaple is the hardest location for the classifier). At Crinnis, the segmentation blockage scores (SegViT and ResNet50-UperNet) are able to capture the progressive increase in blockage of the trash screen, followed by its cleaning. This occurrence was confirmed by visual inspection of images of the screen. However, the classification approach produces blockage scores which do not make a distinction between the fully blocked screen and its progressive, partially blocked, evolution. At Mevagissey, which is a screen placed on a tidal river, the segmentation networks correctly capture the progressive increase and decrease of trash related to part of the debris being flushed away at low tide, which the binary classifier is not able to do. At Siston, the evolution of the trash screen blockage again matches what we could visually see happening at that location. The classification blockage score never goes down to 0 as it did at the other locations. Overall, the results obtained by our segmentation networks are very close and provide the same monitoring trends.

Looking at these results, we conclude that both segmentation networks are able to correctly monitor the temporal evolution of trash screen blockages by monitoring the percentage of the screen that is covered by debris.

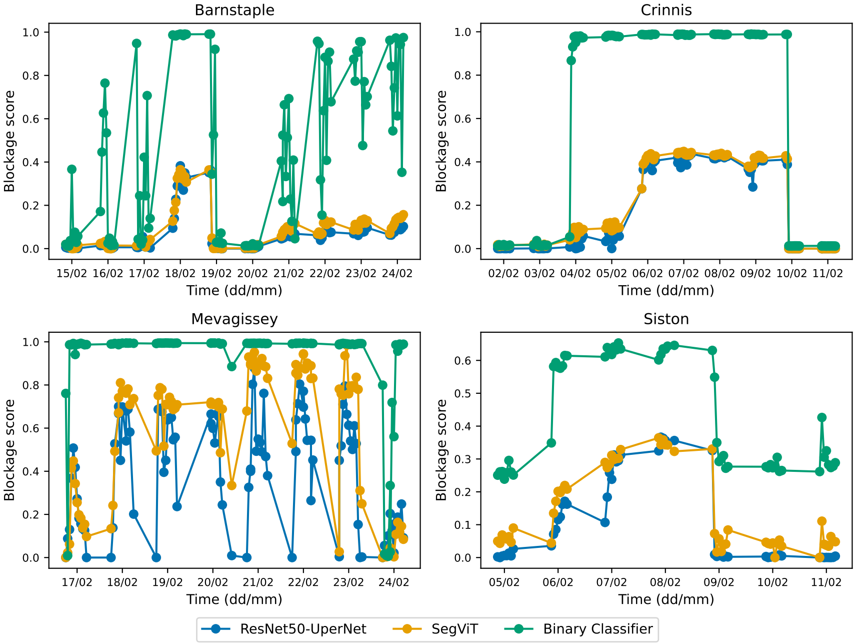

Figure 3: Monitoring trash screen blockage over time using the blockage score. For SegViT and ResNet50-UperNet, the blockage score corresponds to the percentage of the trash screen segmented as debris. For Vandaele et al. [15], the blockage score is the likelihood that the trash screen image is classified as blocked.

# 5 Conclusion

Trash screens are important assets used to prevent debris from entering critical river locations where they could cause damage. However, debris gets stuck on the trash screen and can contribute to floods, so it is important to monitor screens so they can be cleaned effectively. Here we investigate the potential of automatically monitoring trash screens with cameras using deep semantic segmentation networks to provide a pixel-wise mask of the debris location in the images. While cameras already provide useful information, manual analysis is time-consuming and images need to be transformed into actionable information.

In the first part of this study, we created a dataset of trash screen camera images manually labelled with a pixel-wise mask; this dataset is available online [1]. This dataset consists of 575 labelled images augmented to 3950 image crops, coming from 54 different cameras operated by the South West Environment Agency in the UK. These images were originally collated by Vandaele et al. [15].

In the second part of this study, we used this dataset to train two deep debris semantic segmentation networks. The first one, ResNet50-UperNet, is based on a convolutional neural network architecture. The second one, SegViT, is based on a Vision Transformer architecture.

ResNet50-UperNet obtains 68.4 MioU segmentation accuracy on average, while SegViT outperforms it with 77.8 MIoU accuracy. Except for the thinnest debris, we observed that the segmentation obtained with SegViT was of high quality. By using the percentage of the image detected as debris as a blockage score, we compared our work with the classification networks proposed in Vandaele et al. [15] and showed that SegViT was outperforming these methods. We then showed that both segmentation networks could effectively monitor the progressive evolution of trash screen blockages on our test cameras, suggesting a potential application in a warning/alert system.

The promising results of this study demonstrate the potential of deep semantic segmentation networks for trash screen blockage monitoring. Future work will focus on integrating our algorithm into a fully operational platform that offers real-time monitoring. Key areas of development include addressing challenges such as night-time monitoring, implementing an automated alert system, and ensuring the platform can scale effectively for large deployments. By detecting and quantifying blockages, this approach offers a flexible, cost-effective and accurate alternative to human labour for monitoring trash screen blockages in real-time, with potential to help prevent the substantial economic and social impacts of flooding.

# Acknowledgements

This work was in part funded by the Environment Agency under the Flood and coastal resilience innovation programme (FCRIP) and the NERC National Centre for Earth Observation (NCEO). The work has benefited from discussions with members from the FCRIP project *A FAIR (Flood: Aware, Informed, Resilient) approach to community Flood Risk* including local councils from Staffordshire, Stoke and the Black Country as well as JBA Consulting.

# References

[1] Anonymized Author. Automated trash screen blockage segmentation using deep learning: Dataset, 2024. URL https://figshare.com/s/3eb79c15453748f93834.

[2] Jeremy Benn, Barry Hankin, Amanda Kitchen, Rob Lamb, Zora van Leeuwen, and Paul Sayers. *Blockage management guide*. Environment Agency, 2019. URL https://assets.publishing.service.gov.uk/media/60378f4fd3bf7f03985e1286/Blockage_management_guide_-_report.pdf.

[3] Holger Caesar, Jasper Uijlings, and Vittorio Ferrari. Coco-stuff: Thing and stuff classes in context. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1209–1218, 2018. URL https://doi.org/10.48550/arXiv.1612.03716.

[4] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale, 2021. URL https://doi.org/10.48550/arXiv.2010.11929.

[5] EA. EA, EA Web Cams, 2023. URL http://www.eathorvertonwebcam.org.uk/Webcammenu/EAFrameset.html.

[6] Umair Iqbal, Johan Barthelemy, Wanqing Li, and Pascal Perez. Automating visual blockage classification of culverts with deep learning. *Applied Sciences*, 11(16), 2021. ISSN 2076-3417. URL https://www.doi.org/10.3390/app11167561.

[7] Umair Iqbal, Johan Barthelemy, and Pascal Perez. Prediction of hydraulic blockage at culverts from a single image using deep learning. *Neural Computing and Applications*, 34(23):21101–21117, 2022. ISSN 1433-3058. URL https://doi.org/10.1007/s00521-022-07593-8.

[8] Punitha Jaikumar, Remy Vandaele, and Varun Ojha. Transfer learning for instance segmentation of waste bottles using mask r-cnn algorithm. In *Intelligent Systems Design and Applications*, pages 140–149, Cham, 2021. Springer International Publishing. ISBN 978-3-030-71187-0. URL https://doi.org/10.1007/978-3-030-71187-0_13.

[9] Sylwia Majchrowska, Agnieszka Mikołajczyk, Maria Ferlin, Zuzanna Klawikowska, Marta A Plantykow, Arkadiusz Kwasigroch, and Karol Majek. Deep learning-based waste detection in natural and urban environments. *Waste Management*, 138:274–284, 2022. URL https://doi.org/10.1016/j.wasman.2021.12.001.

[10] Roozbeh Mottaghi, Xianjie Chen, Xiaobai Liu, Nam-Gyu Cho, Seong-Whan Lee, Sanja Fidler, Raquel Urtasun, and Alan Yuille. The role of context for object detection and semantic segmentation in the wild. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2014. URL https://doi.org/10.13140/2.1.2577.6000.

[11] Thanh-Thien Nguyen and Hoang-Loc Tran. An efficient model for floating trash detection based on yolov5s. In *Proceedings of the NAFOSTED Conference on Information and Computer Science (NICS)*, pages 230–234, 2022. URL https://doi.org/10.1109/NICS56915.2022.10013413.

[12] Linda J Speight, Michael D Cranston, Christopher J White, and Laura Kelly. Operational and emerging capabilities for surface water flood forecasting. *Wiley Interdisciplinary Reviews: Water*, 8(3):e1517, 2021. URL https://doi.org/10.1002/wat2.1517.

[13] George Streftaris, NP Wallerstein, Gavin Jarvis Gibson, and Scott Arthur. Modeling probability of blockage at culvert trash screens using bayesian approach. *Journal of Hydraulic Engineering*, 139(7):716–726, 2013. URL https://doi.org/10.1061/(ASCE)HY.1943-7900.0000723.

[14] Mohbat Tharani, Abdul Wahab Amin, Fezan Rasool, Mohammad Maaz, Murtaza Taj, and Abubakar Muhammad. Trash detection on water channels. In *Neural Information Processing*, pages 379–389, Cham, 2021. Springer International Publishing. ISBN 978-3-030-92185-9. URL https://doi.org/10.1007/978-3-030-92185-9_31.

[15] Remy Vandaele, Sarah L. Dance, and Varun Ojha. Deep learning for automated trash screen blockage detection using cameras: Actionable information for flood risk management. *Journal of Hydroinformatics*, 26(4):889–903, 04 2024. ISSN 1464-7141. doi: 10.2166/hydro.2024.013. URL https://doi.org/10.2166/hydro.2024.013.

[16] Nicholas Paul Wallerstein and Scott Arthur. A new method for estimating trash screen blockage extent. In *Proceedings of the Institution of Civil Engineers-Water Management*, volume 166, pages 132–143. Thomas Telford Ltd, 2013. URL https://doi.org/10.1680/wama.11.00055.

[17] Nick Wallerstein, Scott Arthur, and D Sisinngghi. Towards predicting flood risk associated with debris at structures. In *Proceedings of 17th Congress of the Asia and Pacific Division of the International Association for Hydro-Environment Engineering and Research (IAHR-APD)*, 2010.

[18] Tete Xiao, Yingcheng Liu, Bolei Zhou, Yuning Jiang, and Jian Sun. Unified perceptual parsing for scene understanding. In *Proceedings of the European Conference on Computer Vision (ECCV)*, September 2018. URL https://doi.org/10.48550/arXiv.1807.10221.

[19] Bowen Zhang, Zhi Tian, Quan Tang, Xiangxiang Chu, Xiaolin Wei, Chunhua Shen, and Yifan liu. Segvit: Semantic segmentation with plain vision transformers. In S. Koyejo, S. Mohamed, A. Agarwal, D. Belgrave, K. Cho, and A. Oh, editors, *Advances in Neural Information Processing Systems*, volume 35, pages 4971–4982. Curran Associates, Inc., 2022. URL https://doi.org/10.48550/arXiv.2210.05844.

[20] Bolei Zhou, Hang Zhao, Xavier Puig, Tete Xiao, Sanja Fidler, Adela Barriuso, and Antonio Torralba. Semantic understanding of scenes through the ade20k dataset. *International Journal on Computer Vision*, 127:302–321, 2018. URL https://doi.org/10.1007/s11263-018-1140-0.