Oct 20th, 2025

# Open Slurry Tank Detection Model Report

River Deep Mountain AI

**Contents**

# List of Figures

# List of Tables

# BB2Ai: Using AI/ML to detect slurry tanks from satellite images

## 1. Introduction

This model aims to accurately identify and geolocate slurry tanks from aerial/satellite imagery. While leveraging YOLOv8, a state-of-the-art object detection architecture developed by Ultralytics, the focus of this work lies not just in deploying a pretrained model, but in curating a custom, annotated dataset, designing a geographically aware training strategy, and building a fully automated geospatial inference pipeline from scratch. The model was trained using a custom labelled dataset. YOLOv8 was chosen for its balance between detection accuracy and speed making it suitable for large-scale imagery analysis.

## 2. Summary of Data Exploration

### 2.1 Above Ground Storage Tank Dataset

130,000 above ground storage tank (ASTs) imagery was used as initial training dataset. These were provided by previous research developed by Robinson et al. (ref. https://www.nature.com/articles/s41597-023-02780-1). The annotation files in Pascal VOC(XML) format needs to be converted to YOLO format (TXT) to make it compatible with YOLO Object Detection model.

### 2.2 Slurry Tank Dataset

The approach involved in collecting slurry tank imagery is as follows:

- Identifying Locations: Determined existing slurry tank locations using Open Street Map Datasets for England, Denmark and Wales**.** The data was downloaded from https://download.geofabrik.de/**.** A total of 93 locations were identified from England, 79 from Denmark and 101 from Wales
- Downloading High-Resolution Imagery: Imagery tiles at 18 zoom level were fetched to ensure detailed images in png format. This zoom level ensured a tile size of 256×256 pixels and 0.6 to 0.7 meters per pixel (i.e., each pixel represents

~60–70 cm on the ground). This would result in an image patch representing ~150–200 $m^2$ on the ground

- Annotating Slurry Tank: Using Label Studio, (https://labelstud.io/) the slurry tanks were manually labelled, and annotations were exported directly in YOLO format to align with training requirements.


Fig. 1: Labelled Images for slurry tanks

It was observed that the dataset size was very low for effective training. To improve model performance and to prevent overfitting, data augmentation techniques were applied to artificially increase the number of training samples.

### 2.1.1 *Image Augmentation*

The main goals of augmentations were to:

- Compensate for limited ground-truth imagery.
- Improve model robustness to lighting, orientation, and noise.
- Prevent overfitting on limited examples of slurry tanks.
- Introduce geometric and photometric diversity similar to real-world satellite variations.

The following transformations were applied using the Albumentations library (python module), each with associated probabilities to simulate realistic variability:

- Geometric Augmentations:
  - Horizontal & Vertical Flip: Simulates different orientations of tanks in satellite images.
  - Affine Transform: Includes small rotations (±15°), zooms (90–110%), and translations (±5%).
- Photometric Augmentations:
  - Brightness & Contrast Shifts: Simulates varying sunlight and shadow.
  - Hue, Saturation, Value Shifts: Captures colour spectrum variability.
  - Gaussian Noise: Models sensor-level noise and compression artifacts.
  - Motion Blur: Mimics slight movement or imperfect captures.
- Artifact Simulations:
  - Image Compression (JPEG noise): Prepares the model for variable imagery quality from web maps or APIs.

Each original image and its corresponding label were used to generate 3 augmented copies, maintaining spatial accuracy of bounding boxes. The benefits included

- Increased dataset size without additional manual annotation
- Improved precision on test images under varied lighting and angles
- Greater model resilience when deployed on real-world imagery with inconsistent quality

Fig. 2: Original image v/s augmented images

### *2.1.2 Image Mosaics for Contextual Diversity*

To further enhance the visual variability and contextual richness of the training dataset, image mosaics were created as part of the augmentation pipeline.

Image mosaicking refers to the technique of combining multiple image tiles into a single composite image. In this project we created multiple mosaic images as shown in Fig. 3. It helps to:

- To introduce spatial context
- To simulate larger scenes and reduce the "tile bias" caused by over-focusing on small areas.
- To increase the effective training data without collecting more raw imagery.
- To improve generalization, especially in detecting slurry tanks located at tile boundaries or near complex rural layouts.

After applying all the techniques mentioned about the total instances of each all the objects have been mentioned in Table 1. Slurry tanks has the class ID 7.

Fig. 3: Mosaic images 2x2,5x5, 6x6

Table 1: Class wise distribution of objects

| Class ID | Class Name | Total Objects |
|---|---|---|
| 0 | closed_roof_tank | 71,826 |
| 1 | external_floating_roof_tank | 10,295 |
| 2 | narrow_closed_roof_tank | 50,125 |
| 3 | sedimentation_tank | 5,657 |

| Class ID | Class Name | Total Objects |
|---|---|---|
| 4 | spherical_tank | 1,684 |
| 5 | undefined_object | 904 |
| 6 | water_tower | 1,330 |
| 7 | slurry_tank | 10,749 |

## 3. Selection of Candidate Features

Slurry tanks are large, typically circular structures used to store liquid waste from agricultural processes. In satellite or aerial imagery, they often appear as:

- Dark or shadowed circular shapes
- Clearly outlined tanks with distinct borders
- Located in rural/farm areas, sometimes near barns or industrial buildings

Unlike traditional machine learning where hand-engineered features are selected, YOLOv8 automatically learns visual features from annotated training data. Therefore, feature selection is replaced by precise and consistent object labelling.

For this model we focused specifically on **open-type slurry tanks**, characterized by:

- Circular shape with visible interior from above
- No overhead structure or roof
- Typically, adjacent to barns or livestock facilities
- Distinct visual contrast (dark waste material, shadow outlines)

Only tanks meeting these criteria were labelled to ensure that the model learned high-quality and consistent visual representations.

## 4. Model Selection

YOLOv8l was selected as the most suitable choice based on its architecture, ease of training, and ability to generalize well on small-to-medium sized objects. YOLOv8 (You Only Look Once, version 8 by Ultralytics is the latest in the YOLO family, offering significant improvements in both detection speed and accuracy. It supports anchor-free object detection and native augmentation, making it well suited for irregular object shapes like slurry tanks.

Since the final objective was to create a GIS-ready output that would scale across multiple geographies, the model needed to process many small image tiles efficiently. YOLOv8's real-time inference capability allowed for:

- Better feature extraction for circular, metallic, or concrete structures.
- Higher precision on test images involving obscure or non-standard tanks.
- Acceptable training times on a T4 GPU using 640x640 image size with a batch size of 8.

## 5. Training and Testing Strategies

To develop a geographically robust object detection model, the training strategy was designed to not only generalize across tank types but also across different countries and landscapes.

### 5.1 Region-Specific Data Split

For the slurry tank class (manually annotated dataset):

- Training Data: Images from England and Denmark
- Validation Data: Entirely separate imagery from Wales

This geographic split was intentional to evaluate the model's ability to generalize beyond the regions it was trained on, a critical capability for large-scale environmental monitoring tools. For the AGST dataset classes, an 70/20/10 random split was used.

### 5.2 Hyperparameter Configuration

Training was carried out using the YOLOv8 large model (yolov8l.pt) with the following optimized settings:

- 300 epochs with early stopping after 20 stagnant epochs
- Image size: 640×640 for higher spatial detail
- Optimizer: AdamW (better stability on mixed datasets), provides more stable convergence on a small and heterogeneous dataset
- Learning rate scheduling from 0.001 to 0.0001
- Weight decay: 0.0005 to prevent overfitting
- 16 data loading workers to speed up GPU pipelines

- Multi-scale training enabled to improve robustness across variable image resolutions
- A mild drop out was included to reduce over fitting, particularly on custom slurry tank annotations

### 5.3 Augmentation Pipeline

Given the relatively limited volume of manually labelled slurry tank images, aggressive data augmentation was used to synthetically expand the dataset and improve generalization:

- Color-based Augmentations:
  - hsv_h: hue shifting (+/- 1.5%)
  - hsv_s: saturation enhancement (up to 70%)
  - hsv_v: brightness variability (up to 40%)
- Geometric Augmentations:
  - Translation: up to 10% horizontal/vertical shift
  - Scaling: up to 50% zoom-in/zoom-out
  - Flips: horizontal (50%) and vertical (10%)
- Advanced Techniques:
  - Mosaic augmentation: randomly merges four training images into one, forcing the model to detect objects in complex contexts
  - MixUp augmentation: blends two images and labels, smoothing decision boundaries
  - Dropout augmentation: randomly drops regions from the image to prevent co-dependency

### 5.4 Training Monitoring and Output

Training outputs were saved in the runs/train/detect directory with per-epoch logs. This includes matrices like

- Classification loss: measuring how well the model classifies each object
- Box regression loss: evaluating the accuracy of bounding box predictions

- mAP and F1-score curves at multiple IoU thresholds

The best-performing model checkpoint was automatically saved based on validation loss or mAP in a form of best.pt file. Benefits of this set-up includes:

- Geographic robustness: Ensures the model works in unseen regions (like Wales)
- Visual resilience: Handles shadowing, low contrast, and shape variation

### 5.5 Image Tiling and Inference Automation

Unlike conventional object detection tasks where inference is run on ordinary RGB images without any spatial context, satellite imagery introduces several unique challenges. Large .tif files must be tiled to fit into GPU memory, and each tile carries spatial referencing metadata that must be preserved. Standard YOLO inference returns bounding boxes in pixel coordinates, but for geospatial applications, those boxes must be converted back to real-world geographic coordinates (latitude and longitude). This requires custom integration of geotransform logic coordinate reprojection (pyproj), and structured export formats like GeoJSON.

To process high-resolution satellite images at scale, a fully automated Python pipeline was developed. This begins by scanning a user-defined directory for large .tif satellite images. Each image is first tiled into 512×512-pixel patches using the rasterio library. This ensures compatibility with GPU memory limits and allows inference to run on smaller, more manageable chunks.

Each tile is then contrast-stretched and converted to PNG format to enhance visual features such as tank edges and shadow contours. Simultaneously, corresponding World Files (.wld) are generated to store geospatial referencing information, preserving the tile's original coordinate system.

Once the tiles are prepared, YOLOv8 inference is performed on the PNG images, and results are automatically saved in both image and YOLO .txt formats. These .txt files contain object class IDs, confidence scores, and normalized bounding box coordinates.

In the final step, a custom post-processing function reads the YOLO detection files and their associated. wld and .tif metadata to reconstruct the bounding boxes in geographic space. Pixel coordinates are transformed into latitude and longitude using affine transformations and CRS reprojection (EPSG:4326). The final output is a GeoJSON file, containing polygon features with class labels, confidence scores, and source image references — ready for use in GIS platforms such as QGIS or ArcGIS.
This modular pipeline ensures efficient, scalable processing of large satellite imagery datasets with automated, georeferenced object detection outputs.

## 6. Evaluation metrics

To assess the performance of our custom dataset trained YOLOv8l object detection model, we evaluated it on a validation set composed of both the pre-annotated AGST dataset and manually labelled slurry tank imagery. Standard object detection metrics were computed using YOLOv8's built-in evaluation pipeline.

The key metrics used include:

- Precision: Measures how many of the detected objects are correct.
- Recall: Measures how many actual objects were detected.
- mAP@0.5: Mean Average Precision with IoU threshold of 0.5, reflecting object-level correctness.
- mAP@0.5:0.95: Averaged mAP over 10 IoU thresholds (0.5 to 0.95), a stricter evaluation of bounding box quality.

Table 2. shown below includes these key metrices for each object class which indicates the model performance. These results confirm that the slurry tank class was one of the best-performing classes, second only to closed_roof_tank in detection robustness. Classes like narrow_closed_roof_tank and undefined_object had very low recall, indicating the model struggled to consistently detect them.

Slurry tanks, on the other hand, maintained a balanced and high performance across all metrics, validating the quality of the custom annotation process and targeted augmentations. The high precision means false alarms are rare, which is critical for GIS use cases where false detections could lead to misinformed environmental decisions.

A recall of ~78% is strong, though slightly lower than other industrial classes, likely due to the visual diversity of slurry tanks and partial occlusion in rural landscapes. The high precision means false alarms are rare, which is critical for GIS use cases where false detections could lead to misinformed environmental decisions.

High mAP@0.5:0.95 reflects the model's ability to place accurate and tight bounding boxes even under strict conditions.

Table 2: Performance Matrices

| **Class** | **Precision** | **Recall** | **mAP@0.5** | **mAP@0.5:0.95** |
|---|---|---|---|---|
| closed_roof_tank | 0.819 | 0.867 | 0.906 | 0.689 |
| external_floating_roof_tank | 0.815 | 0.899 | 0.926 | 0.779 |
| narrow_closed_roof_tank | 0.774 | 0.070 | 0.228 | 0.089 |
| sedimentation_tank | 0.774 | 0.874 | 0.891 | 0.725 |
| spherical_tank | 0.760 | 0.654 | 0.691 | 0.483 |
| undefined_object | 0.278 | 0.145 | 0.129 | 0.110 |
| water_tower | 0.719 | 0.730 | 0.762 | 0.533 |
| **slurry_tank** | **0.945** | **0.778** | **0.883** | **0.737** |

The normalised confusion matrix shown in Fig.4 is a representation of how well each class was predicted relative to the ground truth. The diagonal values represent correct predictions, while off-diagonal values show misclassifications.

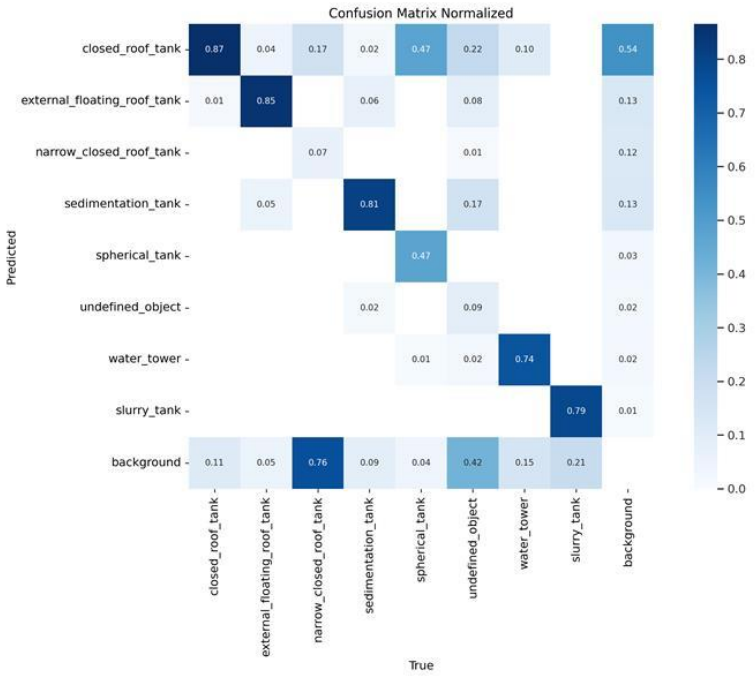


Fig. 4:Normalised confusion matrix

*Performance of slurry_tank (Class 7)*

The model achieved 83% accuracy in correctly classifying slurry_tank objects.

This is one of the highest true positive rates among all classes, indicating the model effectively learned the visual patterns of slurry tanks despite having less training data compared to dominant classes like closed_roof_tank.

Very low confusion with other tanks and minimal misclassification into background or ambiguous categories — showcasing high label purity and model confidence.

This implies that the slurry tank class is well-separated in feature space.

The F1-Confidence curve in Fig.5 shows how F1 score (harmonic mean of precision and recall) changes as the confidence threshold increases. The slurry_tank class maintained a consistently high F1 score across a broad confidence range (~0.2 to 0.8).

It closely follows the top-performing classes like closed_roof_tank, indicating:

- High confidence in detections
- Balanced precision and recall

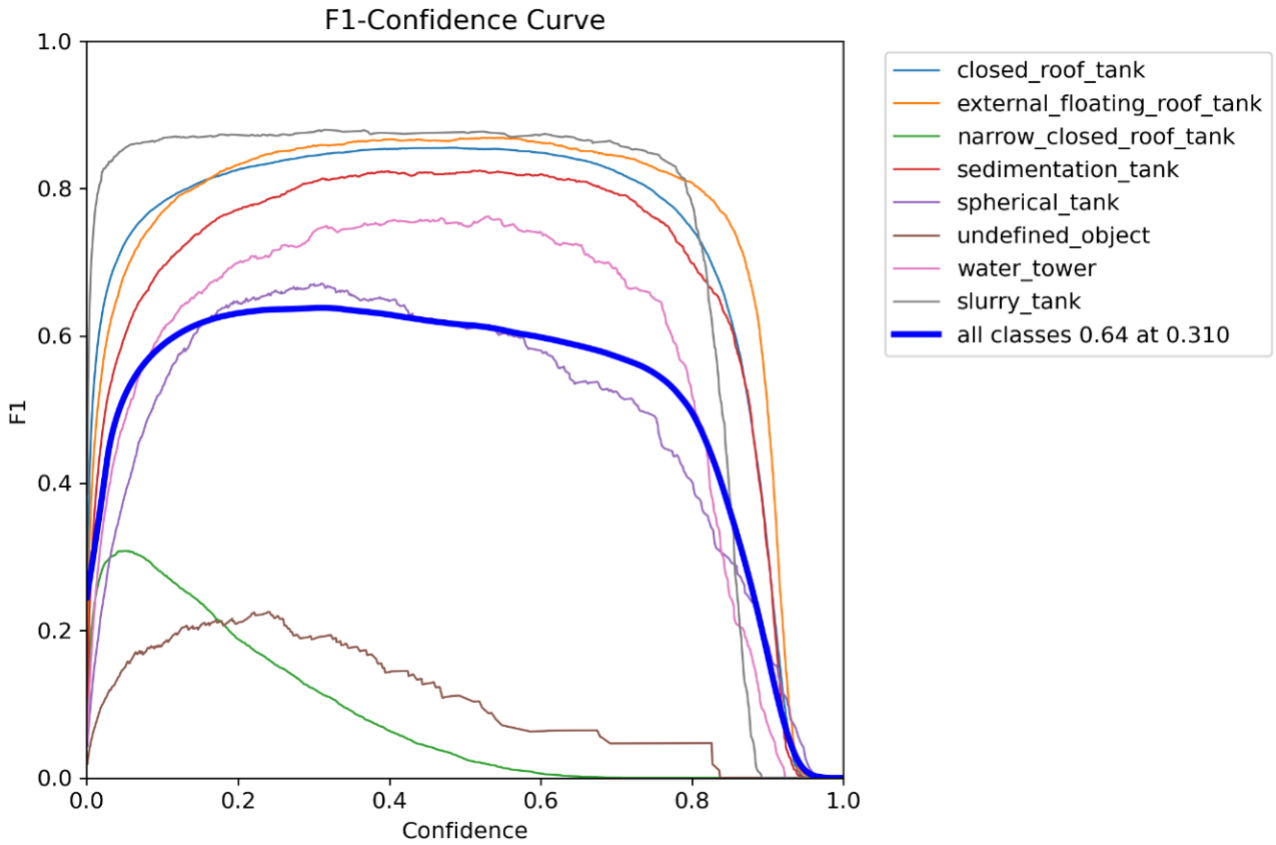


Fig. 5: F1- Confidence curve

The Precision–Confidence Curve in Fig.6 illustrates how the model's precision varies with different confidence thresholds. Each curve shows how likely a predicted object is to be correct (true positive) as we increase the confidence cutoff for detection.

The blue curve represents the averaged precision across all classes, and it reaches a maximum of 1.00 at a confidence.

*Overall Model Behavior:*

- Precision improves as the confidence threshold increases, peaking at nearly 100% at 0.949.
- This suggests that very high-confidence predictions are extremely reliable — but come at the cost of reduced recall (fewer detections).

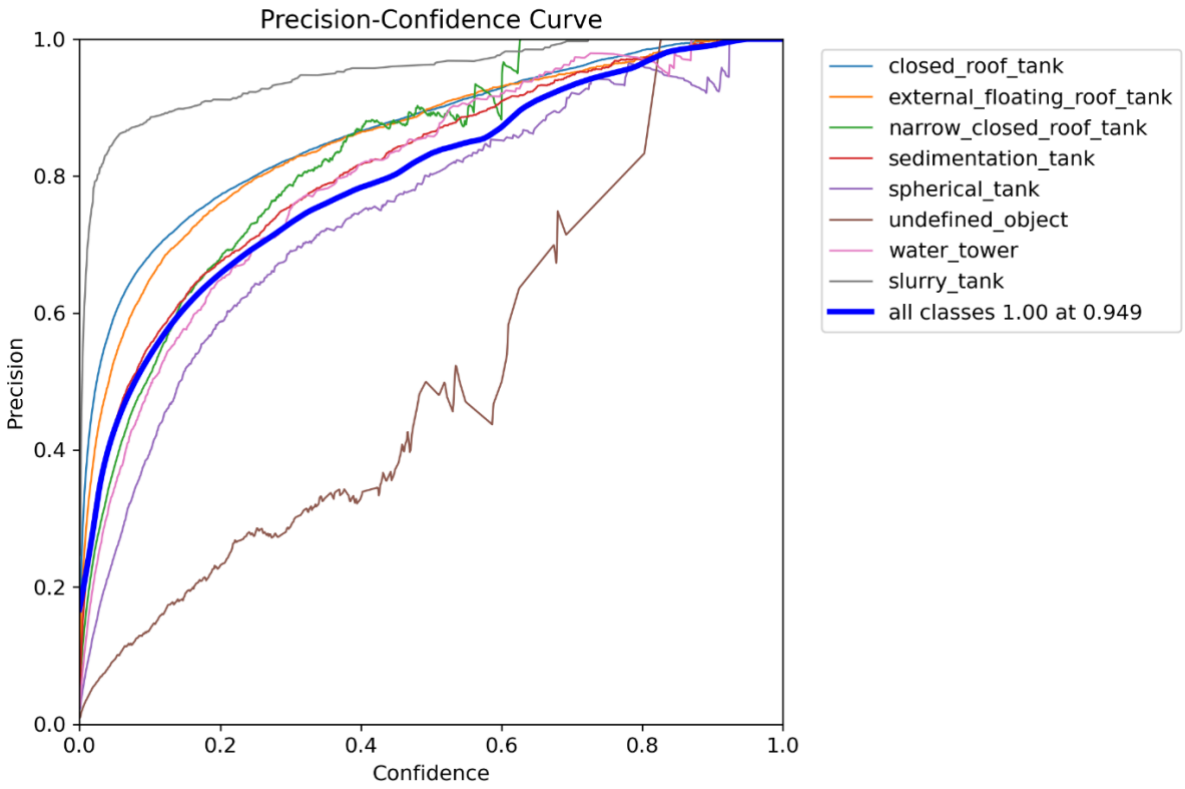


Fig. 6: Precision- Confidence Curve

The Precision–Recall (PR) curve given in Fig.7 provides a detailed view of the model's detection performance for each class. It highlights how well the model balances precision (accuracy of predicted detections) and recall (ability to detect all true objects) at varying confidence thresholds. A higher area under the curve corresponds to better overall class performance, measured here as AP@0.5 (Average Precision at IoU 0.5).

From the curve:

- slurry_tank achieved a high AP@0.5 of 0.883, confirming the effectiveness of the custom annotation and training strategy.
- external_floating_roof_tank (0.926), closed_roof_tank (0.906), and sedimentation_tank (0.891) also demonstrated strong performance, likely due to their distinct shape and strong label representation.
- Conversely, narrow_closed_roof_tank (0.225) and undefined_object (0.128) performed poorly, reflecting either label ambiguity or visual similarity to other tank

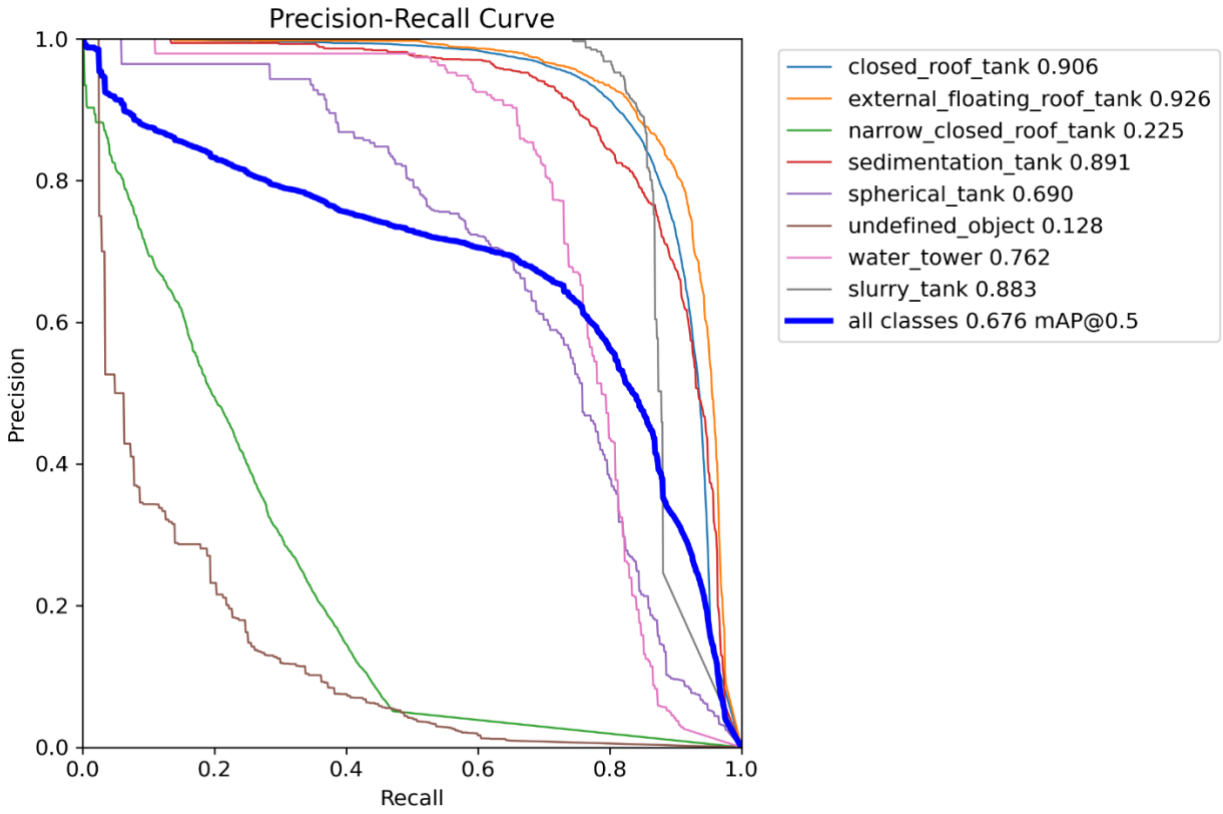


Fig. 7: Precision-Recall Curve

The Recall–Confidence Curve in Fig. 8. illustrates how the model's recall (ability to detect all true positives) varies with different confidence thresholds. High recall at lower confidence thresholds indicates that the model is sensitive enough to detect most objects — even if some are less certain.

- slurry_tank maintains high recall in the 0.2–0.8 confidence range, indicating the model reliably detects these tanks with a reasonable degree of certainty.
- closed_roof_tank, external_floating_roof_tank, and sedimentation_tank achieve consistently high recall, even at moderately high thresholds.
- narrow_closed_roof_tank and undefined_object show sharp recall drops, suggesting either poor generalization or lack of distinctive visual cues

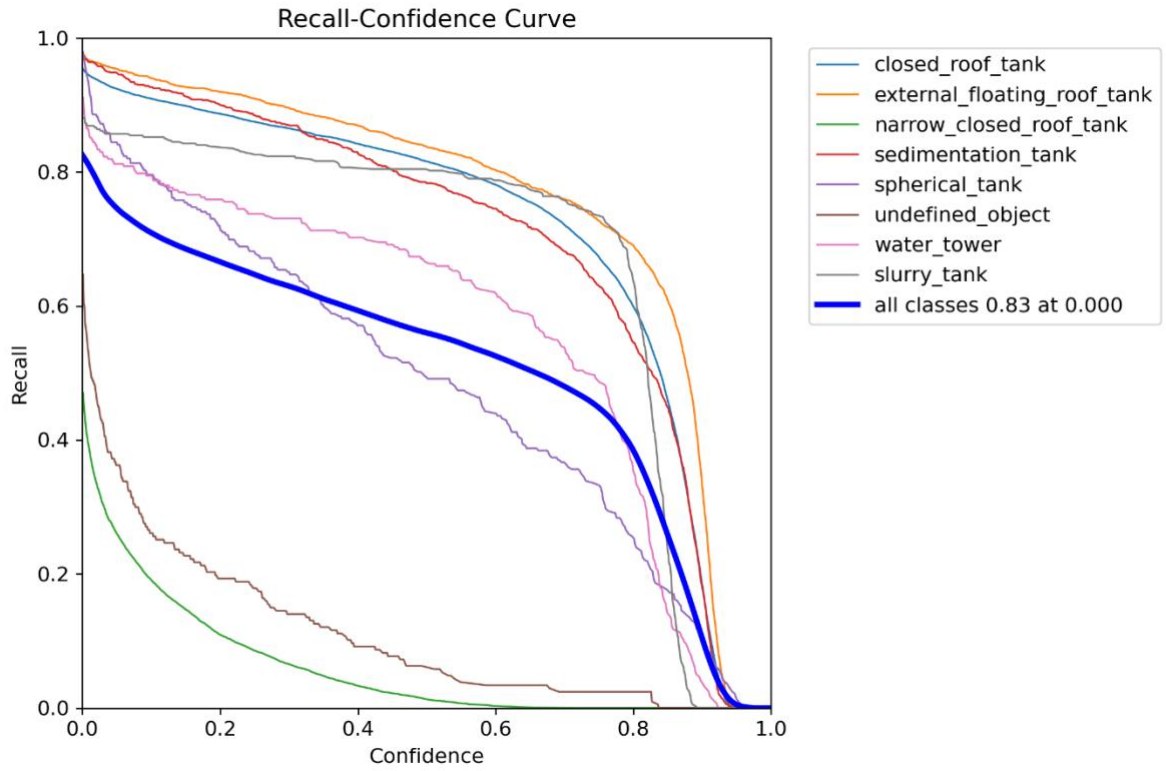


Fig. 8: Recall confidence curve

### 6.1 Takeaways

- Classes with higher structural regularity (e.g., spherical, closed-roof tanks) performed better due to consistent visual features.
- Lower performance in the undefined_object and narrow_closed_roof_tank classes highlights the limitations of both label clarity and training sample diversity.
- Results confirm that the model can be confidently deployed for slurry tank detection in rural settings, especially when imagery has good contrast and clear outlines.

## 7. Phase 2: Validation

Two complementary validation strategies were designed:

**Augmentation-based validation**: Applying flips, rotations, brightness/contrast changes, blur, noise, and compression artefacts on existing images to generate diverse test cases and evaluate robustness.

**Geographic generalisation validation**: testing the trained model on unseen locations to measure performance in areas not represented in the training dataset.

### 7.1 Augmentation-based validation

The main objective of this strategy was to evaluate the robustness of the slurry tank detection model under varying image conditions. Since the dataset consisted of only 270 annotated slurry tank instances, augmentation provided additional validation situations while making efficient use of limited data. The flowchart of the methodology is provided in Fig. 9.

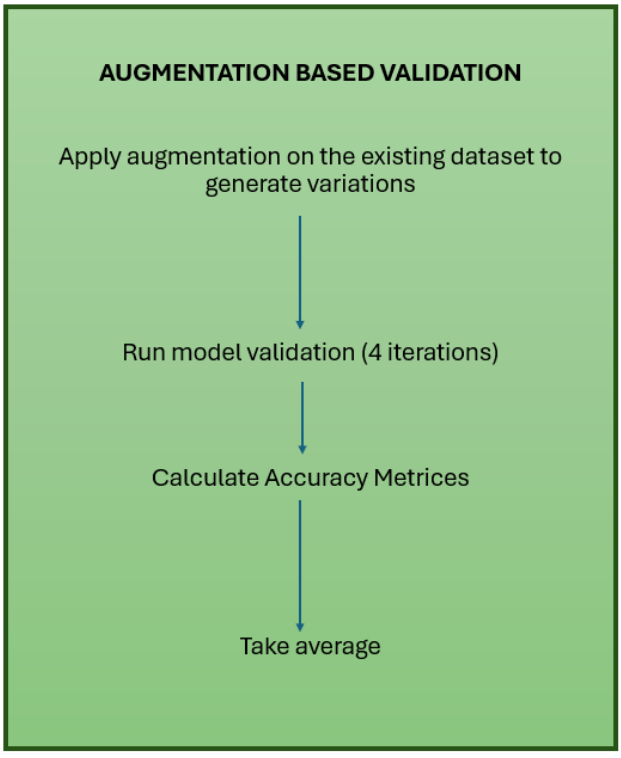


Fig. 9: Flowchart illustrating the augmentation-based validation methodology

### 7.1.1 Methodology

The process began with the original annotated images, which served as the baseline dataset. The AGST dataset contains multiple classes of circular tanks and related structures (e.g.: sedimentation tanks, closed-roof tanks, floating-roof tanks, water towers).

The presence of these non-slurry circular structures makes the dataset valuable for assessing the false positives, where the model might confuse similar looking structures for slurry tanks. This combined dataset design ensured that the validation reflected realistic challenges in distinguishing slurry tanks from visually similar objects in aerial imagery.

For slurry tanks, we combined the annotated images for England, Denmark and Wales. Using the Albumentations library, a series of augmentations (as shown in Table 3) were applied to each image. These augmentations were chosen to represent realistic challenges to interpreting images, such as changes in lighting, orientation, and visual clarity.

For example, transformations included random rotations and flips to mimic different viewing angles, brightness and contrast adjustments to replicate varying environmental or seasonal conditions, and Gaussian noise or blurring to account for differences in image resolution and sensor quality. Colour shifts and sharpening were also applied to evaluate how the model responds to distortions in hue or texture. All augmentations were applied directly on the original, labelled images. Bounding boxes were adjusted automatically by the augmentation library, so the labels always matched the transformed imagery. All augmentations done to images fell within conditions that area observed in aerial/satellite imagery. A short overview of the transformations is provided in Table 3 given below.

Table 3: Augmentations applied to the images

| Category | Operation (Albumentations) | Key parameters | Probability (p) | Notes |
|---|---|---|---|---|
| Geometric | Horizontal flip (HorizontalFlip) | | 0.5 | Applied independently |
| | Vertical flip (VerticalFlip) | | 0.5 | |
| | Affine (Affine) | scale=(0.9,1.1), translate_percent=(0.05,0.05), rotate=(-15,15) | 0.5 | Small shifts/rotations |
| Photometric | Brightness/Contrast (RandomBrightnessContrast) | | 0.3 | Lighting variation |
| | Hue/Saturation/Value (HueSaturationValue) | hue±10, sat±15, val±10 | 0.3 | Colour variation |
| Artefacts | Gaussian noise (GaussNoise) | var_limit=(10,50) | 0.2 | Sensor noise |
| | Motion blur (MotionBlur) | blur_limit=5 | 0.2 | Blur/stitching effect |
| | JPEG compression (ImageCompression) | quality 50–100 | 0.2 | Compression artefacts |

Each augmentation pass was designed to be stochastic: meaning that every time the process was run, a slightly different version of the dataset was generated. Each augmentation operation had a set probability of being applied, and whenever it was applied, its parameters (e.g., rotation angle, brightness shift, noise variance) were randomly sampled from a predefined range. This ensured that the model was not simply being tested on fixed variations but was instead exposed to a wide spectrum of possibilities. All augmentations were applied directly on the original, labelled images and the bounding boxes were adjusted automatically by the augmentation library, so the labels always matched the transformed imagery.

After each augmented dataset was created, the trained YOLO model was run in inference mode, and predictions were compared against the ground-truth bounding boxes inherited from the original annotations.

By repeating this procedure multiple times with different random seeds, the strategy effectively created a rolling validation environment. The model's performance could then be evaluated across diverse visual conditions without requiring

the collection of new ground-truth imagery. This approach provided a practical way to assess robustness while making efficient use of the available data.

### 7.1.2 Results

The augmentation-based validation results were carried out across four independent iterations; the results are summarised in the Table 4 given below.

Table 4: Results

| **Iteration** | **Precision** | **Recall** | **mAP@50** | **Count (Slurry Tank)** |
|---|---|---|---|---|
| 1 | 0.934 | 0.821 | 0.890 | 806 |
| 2 | 0.929 | 0.780 | 0.868 | 809 |
| 3 | 0.939 | 0.830 | 0.896 | 808 |
| 4 | 0.949 | 0.802 | 0.867 | 807 |
| Average: | 0.937 | 0.808 | 0.880 | |

Across the four iterations, precision remained consistently high, averaging 0.937, which indicates that most detections were correct and that the model was not prone to false positives, even under varying augmentation conditions.

Recall values were slightly lower, with an average of 0.808. This suggests that while the model was reliable in its detections, there were instances where true tanks were missed, particularly under more aggressive augmentations such as heavy noise. The mean Average Precision (mAP@50), a more balanced metric, averaged 0.880, confirming strong overall detection performance. Fig. 10 and Fig. 11 represents the normalised confusion matrices for iteration 1 and 2. Each row represents the true class of an object, while each column shows the class predicted by the model. The diagonal values correspond to correct predictions, while off-diagonal values indicate misclassifications.

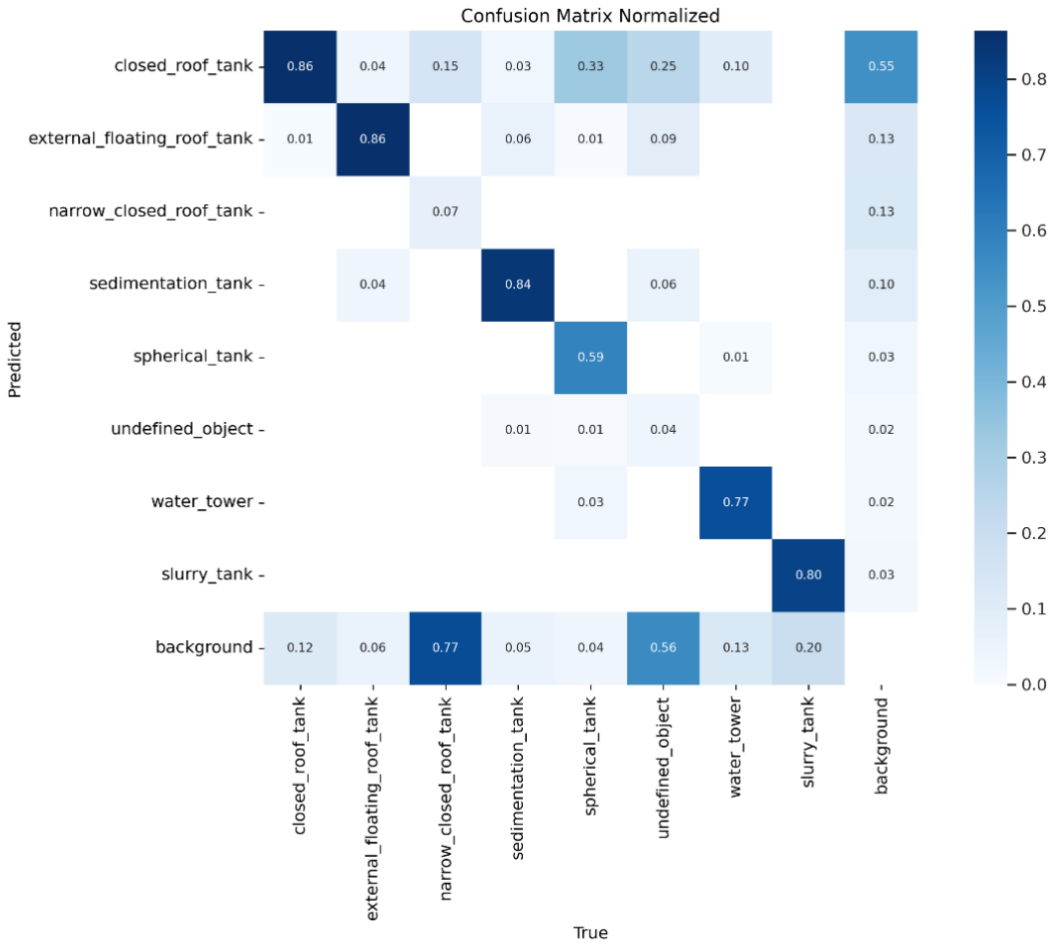


Fig. 10: Normalised confusion matrix for iteration 1

From the matrix, slurry tanks were detected reliably, with most predictions on the diagonal. A small proportion was misclassified as “background”. Some confusion is visible between spherical tanks and closed-roof tanks. This reflects their close visual similarity in aerial imagery.

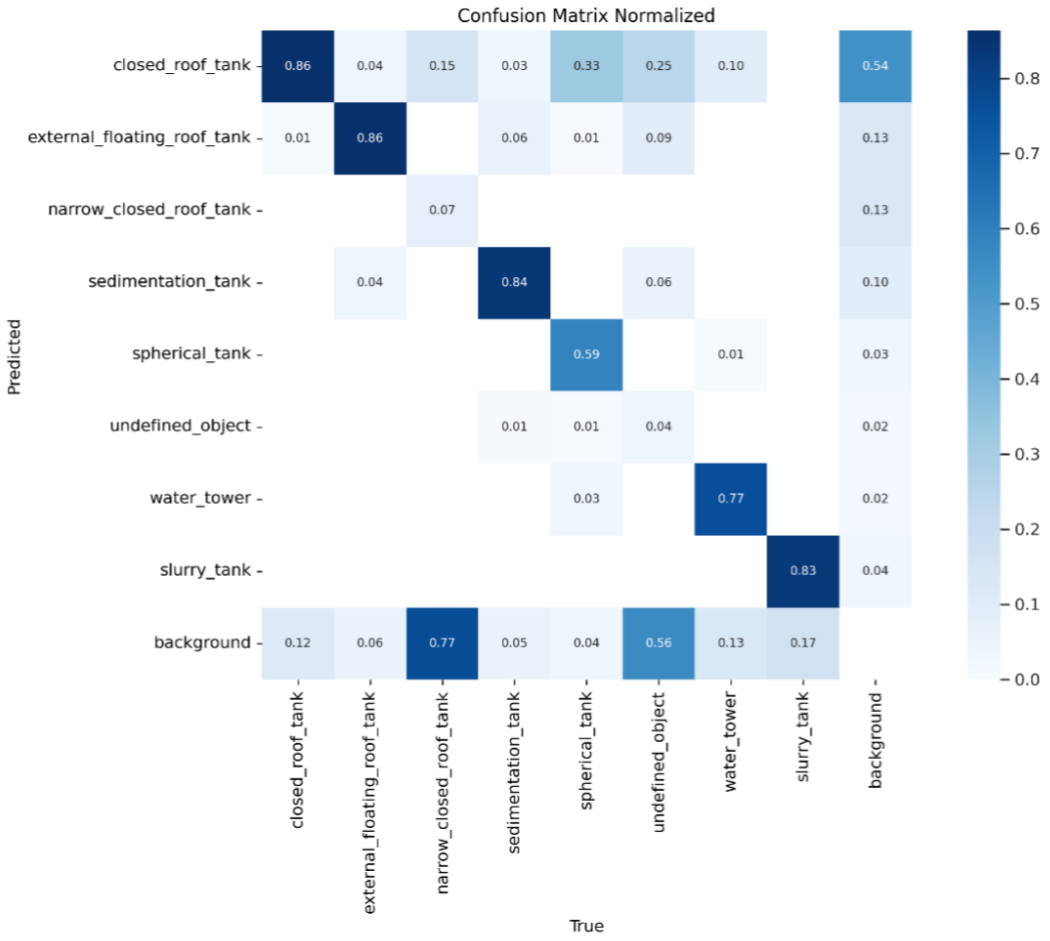


Fig. 11: Normalised confusion matrix for iteration 2

Overall, the matrix demonstrates that the model retains strong performance across the classes.

### 7.2 Geographic generalisation validation

The aim of this validation strategy was to test how well the slurry tank detection model generalises to unseen geographic locations, especially where background conditions or object similarities could challenge the model. Unlike augmentation-based validation, which stresses the model with visual variations of the same dataset, this approach directly evaluates performance in new environments not present during training. The flowchart of the methodology is provided in Fig 12.

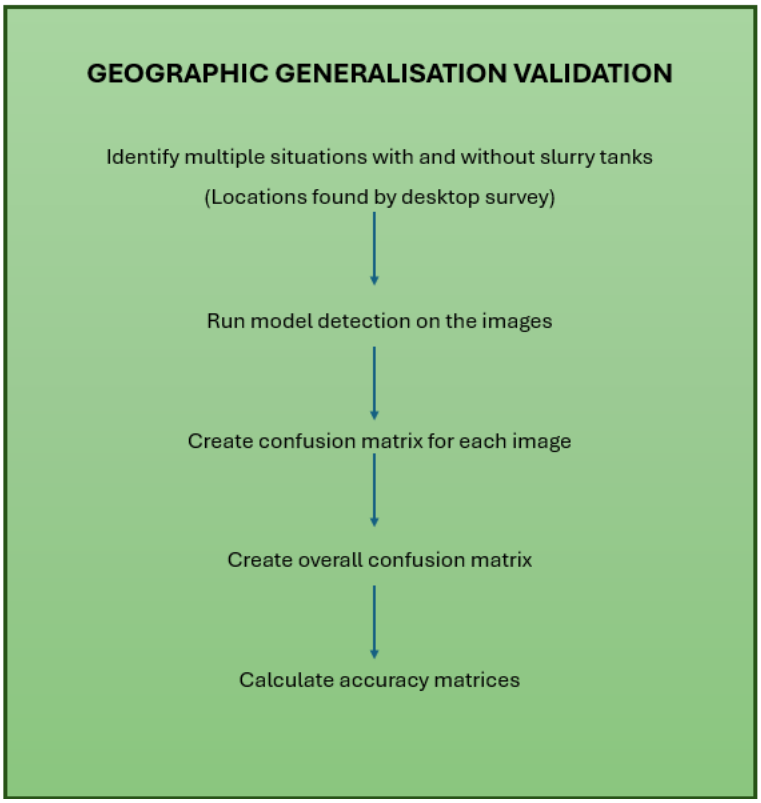


Fig. 12: Flowchart illustrating the geographic generalisation validation methodology

### *7.2.1 Methodology*

A set of diverse geographic locations was carefully selected using a desktop survey. Each location was chosen to represent a particular challenge the model would encounter in deployment. Broadly, the validation was structured around two complementary objectives:

- To evaluate the ability of the model to detect slurry tanks correctly, thereby identifying true positives (TP) and false negatives (FN).
- To test the model's tendency to misidentify other objects as slurry tanks, thereby identifying false positives (FP) and true negatives (TN).

A summary of the instances considered is provided in Table 5 below. A detailed explanation is provided in the sections below. The number of locations selected has also been provided in the table.

Table 5: Summary of Instances Considered

| Sl No. | Instances | No: of locations |
|---|---|---|
| 1 | Farmyards with slurry tanks | 96 |
| 2 | Farmyards with partially hidden slurry tanks | 13 |
| 3 | Farmyards with circular objects but no slurry tanks | 22 |
| 4 | Farmyards with no slurry tanks | 50 |
| 5 | Circular objects outside farmyards | 23 |
| 6 | Circular objects in agricultural landscapes | 12 |
| 7 | Dense clusters of non-slurry tank circular objects | 17 |

The farmyards layer from the Open Street Map dataset for the UK was used to create the validation dataset. Using this layer, a manual desktop survey was conducted. Each farmyard polygon was carefully inspected to identify and categorise sites into relevant validation classes. This survey enabled us to distinguish between:

1. **Farmyards with slurry tanks**: 96 locations were selected where slurry tanks were clearly visible within the farmyard boundaries. These formed the primary basis for identifying True Positives (TP) and False Negatives (FN).
2. **Farmyards with partially hidden slurry tanks:** 13 locations were identified where slurry tanks were present but partially hidden by structures, vegetation, or shadowing. These cases tested the model's ability to detect slurry tanks under incomplete visibility, an important real-world challenge. Examples are given in Fig. 13.

Fig. 13: Examples of partially hidden slurry tanks

3. **Farmyards with circular objects but no slurry tanks:** 22 locations contained objects such as water towers, silos, or other circular structures, which are visually similar and of similar scale (size) to slurry tanks. These sites were critical for detecting False Positives (FP) where the model confuses non-tank objects for slurry tanks. Examples are given in Fig. 14.

Fig. 14: Examples of farmyards with circular objects but no slurry tanks

4. **Farmyards with no slurry tanks:** 50 locations were confirmed via desktop survey to be negative sites, containing no slurry tanks. These served to validate True Negatives (TN) and to ensure the model does not over-predict instances of slurry tanks. Examples are given in Fig. 15.

Fig. 15: Example of farmyards with no slurry tanks

Beyond these farmyard-based categories, additional instances were also considered to challenge the model under more complex generalisation tests:

5. **Circular objects outside farmyards:** 23 locations were selected to test whether detections were overly dependent on spatial context. An example is given in Fig. 16.

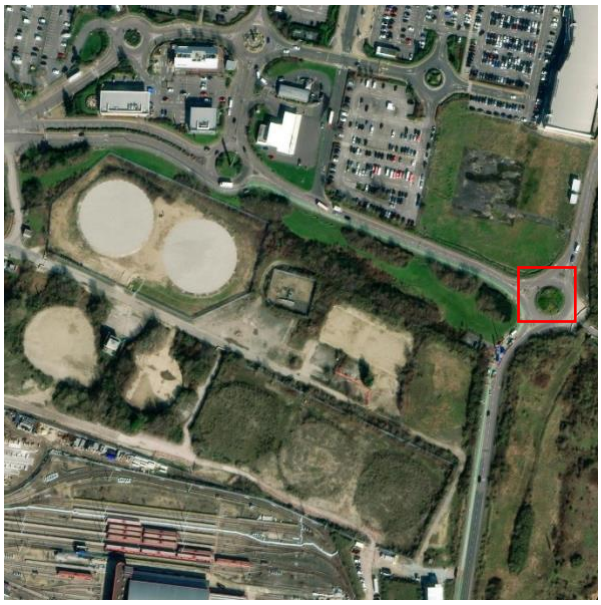

Fig. 16: Example of circular objects outside farmyards

6. **Circular objects in agricultural landscapes (non-industrial settings)**: 12 locations with non-slurry tanks were selected where visual blending could cause confusion. Examples are given in Fig. 17.

Fig. 17: Examples of non-slurry tank circular objects in agricultural landscapes

7. **Dense clusters of non-slurry tank circular objects:** 17 locations were selected to stress-test the model's resilience to false positives in visually cluttered settings. Examples are given in Fig. 18.

Fig. 18: Examples of dense clusters of non-slurry tank circular objects

### *7.2.2 Calculation Approach*

To derive performance metrics, each prediction made by the model was compared against the manually established ground truth at the location level. The outcome of every comparison was classified into one of four categories:

1. True Positive (TP): A slurry tank was present in the imagery, and the model correctly detected it.
2. False Positive (FP): The model detected a slurry tank where none existed (for example, misidentifying a circular water tank or silo as a slurry tank).
3. False Negative (FN): A slurry tank was present but missed by the model.
4. True Negative (TN): No slurry tank was present, and the model made no prediction.

An example of this process is shown in the Fig. 19 below. In the highlighted farmyard, three circular structures are visible. The model predicted one is a slurry tank (a TP) (outlined in yellow), while the other two are not.


Fig. 19: Slurry tank predicted correctly; an example of a True Positive (TP)

In the above example, the model predicted one slurry tank correctly (a TP) and avoided classifying the two non-slurry objects as slurry tanks [avoided False Positives (FPs), achieved True Negatives (TNs)], with the outcomes of this example shown in Table 6.

Table 6: Example Confusion Matrix for the case of a single slurry tank and two non-slurry tank circular objects

| | | PREDICTED | |
|---|---|---|---|
| | | **Slurry Tank** | **No Slurry Tank** |
| **ACTUAL** | **Slurry Tank** | 1 | 0 |
| | **No Slurry Tank** | 0 | 2 |

The example in Fig. 20 shows the instance of a partially hidden slurry tank with no non-slurry tank circular objects. Table 6 shows the resulting confusion matrix when the model was validated against this example - the model correctly identified the single partially hidden slurry tank shown in Fig. 20 and avoided false positives.


Fig. 20: Example of a partially hidden slurry tank and no non-slurry tank circular objects

Table 7: Example Confusion Matrix for the instance of a single, partially hidden slurry tank and no non-slurry tank circular objects

| | | PREDICTED | |
|---|---|---|---|
| | | **Slurry Tank** | **No Slurry Tank** |
| **ACTUAL** | **Slurry Tank** | 1 | 0 |
| | **No Slurry Tank** | 0 | 0 |

The example shown in Fig. 21 illustrates seven non-slurry tank circular structures. The confusion matrix generated from validating the model against this example (Table 7) shows that the model incorrectly identified five out of these seven circular objects as slurry tanks, i.e., 5 False Positives (FPs) and 2 True Negatives (TNs).


Fig. 21: Multiple False Positives in an image

Table 8:Confusion Matrix resulting from validating the model against an instance with 7 non-slurry tank circular objects.

| | | PREDICTED | |
|---|---|---|---|
| | | **Slurry Tank** | **No Slurry Tank** |
| **ACTUAL** | **Slurry Tank** | 1 | 0 |
| | **No Slurry Tank** | 5 | 2 |

In the example shown in Fig. 22, there are many non-slurry tank circular structures. When validating the model against this example, the resulting confusion matric (Table 8) shows that the model falsely predicted only one of these objects (highlighted in yellow) as a slurry tank (False Positive), while the remaining 52 circular objects were correctly not detected as slurry tanks (True Negatives).

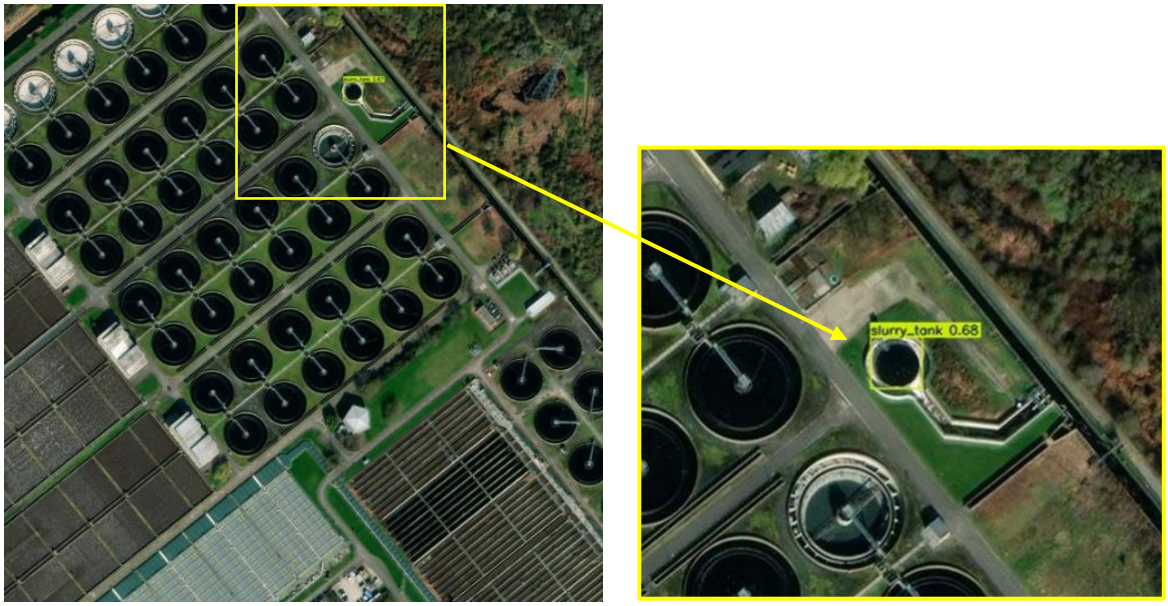


Fig. 22: An example illustrating a dense region of non-slurry tank circular objects

Table 9: Confusion Matrix resulting from validating the model against an example of a dense region of non-slurry tank circular structures, with one False Positive.

| | | PREDICTED | |
|---|---|---|---|
| | | **Slurry Tank** | **No Slurry Tank** |
| **ACTUAL** | **Slurry Tank** | 0 | 0 |
| | **No Slurry Tank** | 1 | 52 |

### *7.2.3 Results*

For each validation instance, a confusion matrix was generated at the image level to record the number of true positives, false positives, false negatives, and true negatives. These individual results were then aggregated across all instances to produce a consolidated confusion matrix, as shown below in Table 10.

Table 10: Overall Confusion Matrix

| | | PREDICTED | |
|---|---|---|---|
| | | **Slurry Tank** | **No Slurry Tank** |
| **ACTUAL** | **Slurry Tank** | 113 | 2 |
| | **No Slurry Tank** | 46 | 645 |

From the aggregated confusion matrix, the standard performance metrics were computed using the following formulas:

- Precision = TP/(TP+FP)
- Recall = TP/(TP+FN)
- F1 Score = 2*Precision*Recall/(Precision+Recall)

The calculated performance matrices are:

- **Precision: 71.0%** - This means that when the model predicted an object as slurry tank, it was correct about 7 out of 10 times.

- **Recall: 98.26%** -confirming that almost all true slurry tanks were successfully detected.
- **F1 Score: 82.48%** -providing a balanced measure of detection reliability and completeness.

These results suggest that the model is highly sensitive (very unlikely to miss a slurry tank), as demonstrated by the near-perfect recall. However, the precision is comparatively lower, reflecting the model's tendency to generate false positives, particularly in environments with visually similar circular structures.

This trade-off is important: in many real-world applications: a higher recall is desirable, as it ensures that potential slurry tanks are not overlooked. False positives can then be reduced through manual review or rule-based filtering.

The accuracy is defined as:

$$\textbf{Accuracy} = (TP+TN)/(TP+FP+FN+\textbf{TN})$$

In object detection, however, the number of True Negatives (TN) is not well-defined. Unlike classification problems (where every sample is either correctly or incorrectly classified), detection models like YOLO scan through thousands of possible bounding boxes per image. Most of these are background regions (not tanks) and correctly rejecting them **would artificially inflate TN as seen in Table 8**, leading to misleadingly high accuracy values. In this validation, TN was used only in a controlled sense: when an entire validation location (e.g., a farmyard or industrial site) contained no slurry tanks, and the model correctly made no detections. This interpretation is practical for situation-based validation.

Because of this, accuracy is not a meaningful measure in object detection, and it is rarely reported. Instead, metrics like Precision, Recall, and F1 are used, as they focus directly on the detection task.

It is also acknowledged that the composition of the validation dataset can influence the resulting confusion matrices, since the relative balance of slurry tanks and non-slurry circular objects affects the distribution of true and false detections. While

catchment-level validation would provide the most representative assessment, no suitable catchment with a sufficient number of known slurry tank locations was identified at this stage. Instead, a structured dataset was constructed from the OpenStreetMap farmyards layer and other circular features, with manual inspection and categorisation into relevant validation classes. This approach ensured that the validation remained systematic and reasonably representative, even if not fully catchment based.

## 8. Conclusions

This project successfully developed and validated a deep-learning pipeline for detecting slurry tanks and similar industrial structures from high-resolution aerial and satellite imagery. Using YOLOv8l as the core detection engine, the model achieved high precision, especially on the custom-annotated slurry_tank class, demonstrating its effectiveness for practical geospatial applications.

Through the combination of:

- A well-curated hybrid dataset (public + manually labelled),
- robust augmentation strategies,
- and geospatial post-processing using .wld files and GeoTIFF metadata,
- the pipeline outputs GIS-ready polygon features that can be directly visualised and analysed in tools like QGIS and ArcGIS.

**Key Strengths:**

- High accuracy on the target slurry tank class
- End-to-end automation from raw .tif imagery to georeferenced. geojson output
- Scalability for processing large tile-based imagery archives
- Modular design, allowing future adaptation for other infrastructure types

**Limitations:**

- Lower recall on narrow or partially occluded tanks due to visual ambiguity

- Imbalance in class representation affects generalisation on underrepresented structures
- Current pipeline supports only RGB imagery

**Future Development**

- The current model only identifies above ground circular slurry tanks, but these only account for roughly a quarter of manure storage systems. The model could be further expanded to consider below ground tanks, slurry lagoons and other potential agricultural pollutant point sources such as silage clamps
- The outputs of this model could be incorporated into the Risk Model developed in RDMAI, particularly If the additional manure storage systems were included.

For internal purposes, this model was trained primarily on high-resolution imagery obtained from the zoom level 18, which typically corresponds to a ground sampling distance (GSD) of ~0.5 to 1 meter per pixel depending on the region. As a result, the visual features learned by the model, such as tank edges, shadows, textures, and contextual cues like farm boundaries, are tightly coupled to this level of spatial detail.

Implication: The model performs best on satellite or aerial imagery with similar spatial resolution and spectral characteristics to the tiles used during training.

## References

1. Robinson, C., Bradbury, K., & Borsuk, M. E. (2024). Remotely sensed above-ground storage tank dataset for object detection and infrastructure assessment. *Scientific Data, 11*, 67. https://doi.org/10.1038/s41597-023-02780-1

2. Khalili, B., & Smyth, A. W. (2024). SOD-YOLOv8—Enhancing YOLOv8 for small object detection in aerial imagery and traffic scenes. *Sensors, 24*(19), 6209. https://doi.org/10.3390/s24196209

3. Wang, X., Qian, H., Xie, L., Wang, X., & Li, B. (2024). Recognition and classification of typical building shapes based on YOLO object detection models. *ISPRS International Journal of Geo-Information, 13*(12), 433. https://doi.org/10.3390/ijgi13120433

4. Ultralytics. (2024). *YOLOv8 model documentation*. https://docs.ultralytics.com/models/yolov8/