GEO/PROJ/22/GIUP
Version C1.0
31 March 2023

Urban Habitat and Naturalness Mapping Phase 1 Roll Out – User Guide

Conducted and Delivered by

(a 2Excel Company)

dated 31 March 2023

Version History

Revision Date	Amendments	Issued to
31/03/23	First Submission – Ver C1.0	Martin Moss

Approvals

Prepared by .... Loreena Jaouen BSc MSc
Geospatial Product Development Lead

.... Klara Halikova BSc
Senior Remote Sensing and Geospatial Analyst

Reviewed by .... Chloe Barnes BSc MSc PhD
Head of Remote Sensing

References

1. Urban Habitat and Naturalness Mapping Phase 1 Roll Out – Technical Report Ver C1.0 [Submitted on 31/03/23 to Martin Moss]

Acronyms

Acronym	Meaning
AOI	Area Of Interest
APGB	Aerial Photography for Great Britain
BI	Blue Infrastructure
BNG	British National Grid
BUA	Built-Up Areas
CIR	Colour Infra-Red (Red, Green, Near Infra-Red)
CRS	Coordinate Reference System
DSM	Digital Surface Model
DTM	Digital Terrain Model
EA	Environment Agency
GI	Green Infrastructure
ML	Machine Learning
NFI	National Forest Inventory
NIR	Near Infra-Red
OS	Ordnance Survey
OSMM	Ordnance Survey Master Map
PHI	Priority Habitat Inventory
RGB	Red, Green, Blue (true colour)

Introduction

This document is intended as a step-by-step user manual and should allow anyone with access to the required software (**Section 2**) and datasets (**Section 3**) to reproduce the outputs from the Urban Habitat and Naturalness Mapping Phase 1 Roll Out project [**Ref 1**] for their chosen urban area of England. An overview of the workflow is shown in **Figure 1** and the location of all supporting files provided by 2Excel is summarised in **Table 1**.

This guide assumes the use of the England Green Infrastructure Mapping Database Version 1.2 (issued on 31/01/2023).

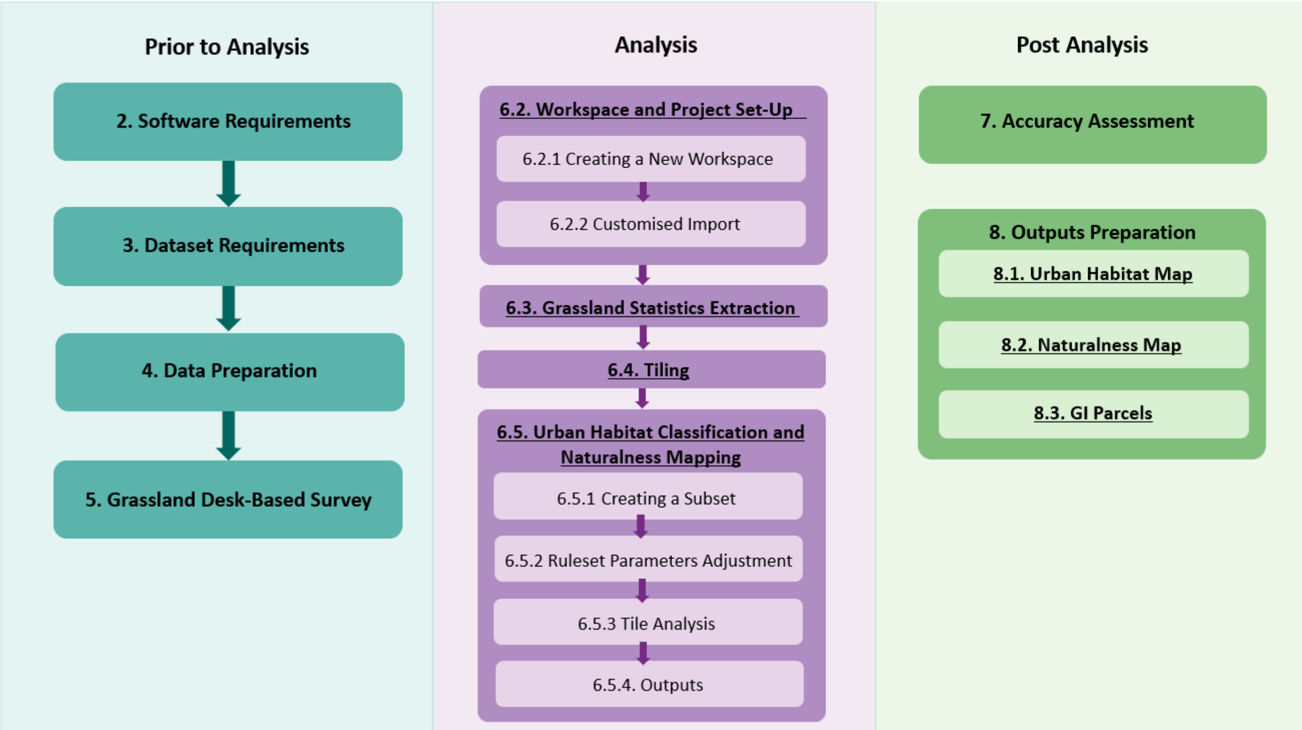


Figure 1. Summary diagram of the workflow.

Guide Section		Required Files	Location
4. Data Preparation	4.4. APGB	prepare_apgb_by_date.py	*Supporting Files*\python_scripts
	4.5. AOI	create_apgb_date_blocks.py	
	4.8. EA LiDAR	prepare_ea_lidar.py	
5. Grassland Desk-Based Survey	5.1. Survey Set-up	Example_Shapefile_Template - Grassland_Survey.shp	*Supporting Files*\QGIS\ Grassland Desk-Based Survey
		Grassland_Samples_Styling.qml	
		Broad_value_map_grasslands.csv	
		Detailed_value_map_grasslands.csv	
		GI_Classes.qml	
6. Analysis Workflow	6.2. Workspace and Project Set-Up	GI_import.xml	*Supporting Files*\eCognition\imports
	6.3. Grassland Statistics Extraction	GI_0_grass_stats.dcp	*Supporting Files*\eCognition\rulesets
	6.4. Tiling	GI_1_tiling.dcp	
	6.5. Urban Habitat Classification and Naturalness Mapping	GI_2_classification.dcp	
7. Accuracy Assessment	7.1. Project Set-up	Detailed_Classes_for_Assessment_Styling.qml	*Supporting Files*\QGIS\ Accuracy Assessment
		Broad_value_map.csv	
		Detailed_value_map(v2).csv	
		GI_Classes.qml	
		NFI_Classes.qml	
		PHI_Classes.qml	
		OSMM_Buildings.qml	
		OSMM_Gardens.qml	
		CHM.qml	
	7.3. Confusion Matrix	generate_confusion_matrix.py	*Supporting Files*\python_scripts
8. Outputs Preparation	8.1. Urban Habitat Map	rename_gdb_tiles.py	*Supporting Files*\python_scripts
	8.3. GI Parcels	calculate_combined_naturalness.py	

Table 1. Location of supporting files needed in the workflow, where *Supporting Files* corresponds to the folder provided by 2Excel.

1. Urban Habitat Classification Scheme

A summary of the urban habitat classification scheme used in the roll-out project [**Ref 1**] can be found in **Table 2**. The classification scheme contains 11 broad classes and 30 detailed classes, each with an associated key. Some broad classes, such as Private Gardens (F) and Parklands (H), are contextual and may contain features that can also be found in other broad classes, e.g., Grasslands (A), Woodlands (B) and Rough, Abandoned and Derelict Land (C). The distinction between woodlands and scattered trees is based on surface area, where woodlands are defined as any cluster of trees equal to or larger than 0.5 hectares.

Broad Key	Broad Class Name	Detailed Key	Detailed Class Name
A	Grasslands	A1	Amenity Grassland
		A2	Undifferentiated Grassland
B	Woodlands	B1	Broadleaved, Mixed and Yew Woodland
		B2	Conifer-Dominated Woodland
		B3	Isolated and Scattered Trees
C	Rough, Abandoned and Derelict Land	C1	Habitat Mosaics
		C2	Scrubs
D	Wetlands	D1	Open Water
		D2	Vegetated Wetland
E	Impervious and Non-Vegetated	E1	Sealed Surfaces and Buildings
		E2	Vegetated Building Surfaces and Green Roofs
		E3	Bareground
F	Private Gardens	F1	Non-Vegetated Gardens
		F2	Vegetated Gardens
		F3	Garden Trees
		F4	Garden Scrubs
G	Formal Planting	G2	Allotments
H	Parklands	H1	Park Amenity Grassland
		H2	Park Undifferentiated Grassland
		H3	Park Wood Pasture
		H4	Park Scrubs
I	Coastal	I1	Coastal Sand
		I2	Coastal Dunes
		I3	Coastal Shingle, Loose and Bare Rocks
		I4	Coastal Mud
		I5	Coastal Saltmarshes
		I6	Coastal Cliffs and Slopes
J	Agricultural Land	J1	Vegetated Fields
		J2	Ploughed Fields
K	Upland Habitats	K1	Upland Habitats

Table 2. Urban Habitat Classification Scheme.

2. Software Requirements

2.1. QGIS 3.22 [open source]

Download: https://www.qgis.org/en/site/forusers/download.html

The QGIS software is primarily used to prepare and visualise datasets, and to perform the grasslands desk-based survey and final accuracy assessment. Whilst data preparation and visualisation could be done in ArcGIS (or any other suitable GIS software), the desk-based survey and accuracy assessment steps rely on 2Excel's QGIS styling and survey templates. As a result, QGIS 3.22 is recommended for ease of use[1].

2.2. Python 3.7 [open source]

Download: https://docs.conda.io/en/latest/miniconda.html

Python is primarily used to prepare datasets for mapping, and to finalise the outputs. The scripts used in this project [**Ref 1**] have been written in Python 3.7 and documentation can be found in **Appendix A**, **B**, **C**, **D**, **E** and **F**. They may be transferable to other versions, but this has not been tested.

2.3. LASTools [commercial]

Download: https://rapidlasso.com/lastools/

LASTools is used to merge the EA LiDAR tiles into single files. A free version of the software exists for smaller areas, but artefacts are voluntarily introduced at larger scales when used without a commercial license. Any software that allows LAZ tile merging could be used as an alternative.

2.4. Trimble eCognition 10.3 [commercial]

Download: https://geospatial.trimble.com/ecognition-download

This is the main software used for the urban habitat and Naturalness mapping workflow and no alternative is currently available. Licenses need to be purchased, although a free trial version is available for initial testing and training. The official UK provider for Trimble eCognition software licenses is Korec: https://www.korecgroup.com/.

There are two types of eCognition license: Developer and Server. Developer licenses allow ruleset modifications and at least one is required for initial threshold setting. The Server licenses allow tiling of larger areas for more efficient parallel batch processing. It is advised to purchase as many Server licenses as can be supported by the user's computing power and budget.

[1] Users could adapt the QGIS styling and survey templates to the ESRI suite of software, but this is outside the scope of the User Guide

eCognition licenses can be renewed monthly and offer flexibility. Users should consider preparing their datasets and performing initial tests before purchasing Server license for efficient use of the license during the rental period.

3. Dataset Requirements

All datasets required for the analysis workflow are listed in this section. Whilst some datasets can only be downloaded at national-scale, others require a zone to be specified due to the size of the data, e.g., APGB, OSMM and EA LiDAR. In this case, the AOI plus a 500m buffer (**Section 3.2**) should be used.

3.1. Data Download Directory

Raw, unprepared data refer to datasets as they appear straight after download in their original folder structure and without any modifications. All raw datasets should be stored in a "data download directory" (or "DataDir"). The name and location of "DataDir" can be chosen by the user but should remain consistent across all datasets. Certain datasets will need to follow a strict folder structure within "DataDir" to allow for subsequent automated preparation (e.g., APGB and EA LiDAR), but this will be explained in the relevant section.

During data preparation (**Section 4**), a new directory will be created ("RootDir"). This will contain all prepared datasets (as opposed to raw datasets) that are required for analysis in eCognition. This time, all datasets will follow a strict folder structure, naming convention and format.

3.2. Area Of Interest [defined by user]

The AOI is a vector layer which outlines a zone to constrain the analysis. Everything within the AOI will be mapped, and anything outside of it will be discarded. It is a requirement of the workflow and should be in the ESRI Shapefile format. Whilst the AOI can be obtained from anywhere and sourced by the user, it is recommended to use Local Authority boundaries. These can be extracted from the GI database (GI Social Statistics – map 6) (**Section 3.3**). Multiple Local Authority boundaries can be merged and dissolved into a single AOI for larger metropolitan counties.

Due to the nature of the classification scheme (**Section 1**), the context and surface area of urban habitats matter. To ensure that parcels cut-off by the edge of the AOI are still correctly classified, and to ensure coastal habitats are fully captured, the AOI should be buffered by 500m.

3.3. Green Infrastructure and Blue Infrastructure [Open Government Licence]

Download: https://www.data.gov.uk/green-and-blue-infrastructure-england

The GI and BI geodatabases published by Natural England form the basis of the urban mapping. They are used to provide context to the spectrally informed habitat classification. The Naturalness factor is also attached to each GI parcel (GI Access Maps – map 1) at the end of the workflow. This guide assumes that the England Green Infrastructure Mapping Database Version 1.2 (issued on 31/01/2023) is used. The full national dataset should be downloaded.

3.4. Aerial Photography for Great Britain [Public Sector End User Licence]

Download: https://www.apgb.co.uk/

The APGB dataset provides spectral information and is the main dataset used in the urban habitat and Naturalness mapping workflow. Both RGB (12.5 cm spatial resolution) and CIR (50cm spatial resolution) datasets should be acquired for the AOI 500m buffer (**Section 3.2**), and downloading the latest imagery is recommended. However, the users should make sure that the imagery date of capture is between April and September during the leaf-on season. Any data capture outside of this period will make habitat classification difficult or near impossible. These datasets are delivered as 1km tiles and dates of collection may vary (**Figure 2**). The outlines of the different "date blocks" will be generated in **Section 4.5** using a Python script and will be used in subsequent data preparation steps – this is because each date block will require different spectral thresholds due to changing conditions. Step-by-step instructions on merging APGB tiles by date to create virtual mosaics will be provided in **Section 4.4**.

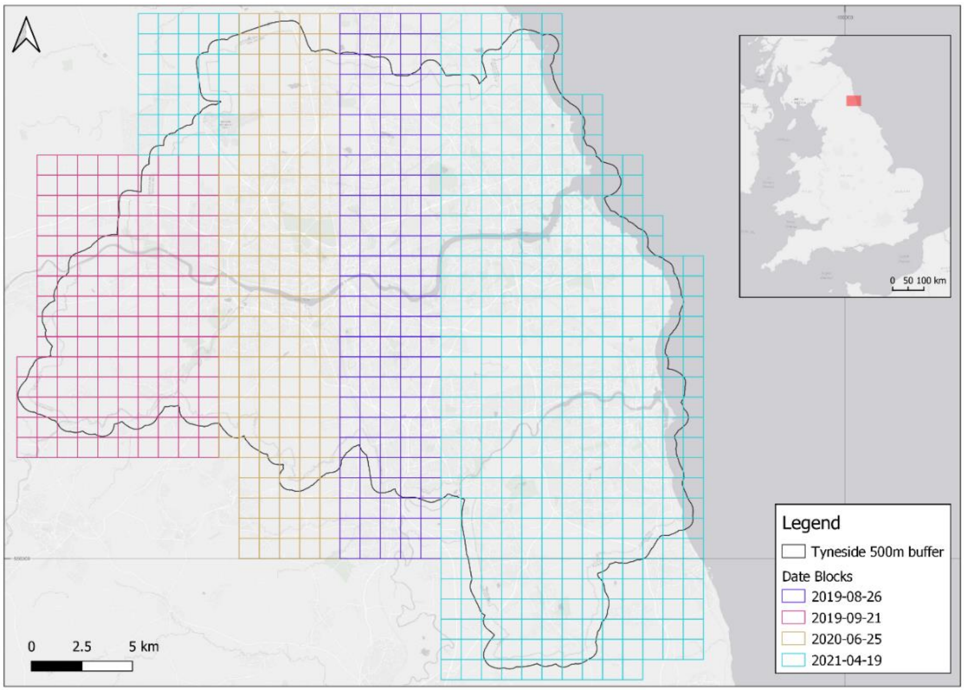


Figure 2. Example of the four APGB dates that make up Tyneside.

The DTM and DSM datasets are optional and are only needed if the date of the EA LiDAR acquisition does not align with the APGB information, or if the user's computing power does not allow for LiDAR point cloud handling.

The downloaded tiles must be placed as they are in an APGB folder within "DataDir", following the structure:

- {DataDir}\APGB\RGB\{aoi}
- {DataDir}\APGB\CIR\{aoi}
- {DataDir}\APGB\DTM\{aoi}
- {DataDir}\APGB\DSM\{aoi}

The *aoi* parameter corresponds to the name of the AOI (usually a Local Authority or Metropolitan County name). If it contains multiple words, spaces should be suppressed and upper-case letters should be used to signify the start of a word, e.g., GreaterManchester or WestMidlands.

3.5. Ordnance Survey British National Grid [Open Government Licence]

Download: https://github.com/OrdnanceSurvey/OS-British-National-Grids

Both 1km and 5km OS grids are required as part of the workflow to create APGB date block outlines and tile projects in eCognition. The full national dataset should be downloaded.

3.6. Ordnance Survey Master Map [Public Sector End User Licence]

Download: https://environment.data.gov.uk/

The OSMM dataset is required to provide private garden information, as well as building footprints. OSMM greenspace information is also used to enhance the classification of certain classes, especially along coasts. It should be downloaded for the AOI 500m buffer (**Section 3.2**)

3.7. National Forest Inventory [Open Government Licence]

Download: https://www.data.gov.uk/national-forest-inventory-woodland-england-2020

The NFI dataset provides information about woodland types. The latest version of the full national dataset should be acquired.

3.8. Environment Agency National LiDAR Programme [Open Government Licence]

Download: https://environment.data.gov.uk/DefraDataDownload/?Mode=survey

Both DTM and point clouds should be acquired for the AOI 500m buffer (**Section 3.2**). The EA LiDAR point cloud is used to generate a higher resolution DSM than that provided by the APGB dataset. These are downloaded as zipped files in 5km tiles. Step-by-step instructions for unzipping and merging of the tiles will be provided in **Section 4.7**. Note that depending on hardware specifications, merged point clouds for certain date blocks may be too large to handle in eCognition and may need to be split up further – more details are provided in **Section 10.2**.

The downloaded tiles must be placed as they are in an EA_LiDAR folder within "DataDir", following the structure:

- {DataDir}\EA_LiDAR\DTM\{aoi}
- {DataDir}\EA_LiDAR\LAZ\{aoi}

The *aoi* parameter corresponds to the name of the AOI (usually a Local Authority or Metropolitan County name). If it contains multiple words, spaces should be suppressed and upper-case letters should be used to signify the start of a word, e.g., GreaterManchester or WestMidlands.

3.9. OS Open Built-Up Areas [Open Government Licence]

Download: https://www.data.gov.uk/os-open-built-up-areas

The BUA dataset is published by the OS and delineates built-up areas in Great Britain. It is used to enhance the mapping of agricultural land. The full national dataset should be downloaded.

3.10. Priority Habitat Inventory [Open Government Licence]

Download: https://www.data.gov.uk/priority-habitats-inventory-england

The PHI dataset published by Natural England provides additional information about coastal habitats, upland habitats and vegetated wetlands that would otherwise be difficult to identify using airborne imagery alone. The full national dataset should be downloaded[2].

3.11. Moorland Line [Public Sector Mapping Agreement End User Licence]

Download: https://magic.defra.gov.uk/Datasets/Dataset_Download_MoorlandLine.htm

This dataset is maintained by the Rural Payments Agency (RPA) and is used to roughly delineate upland areas. The full national dataset should be downloaded.

[2] Note that this dataset comes in two parts which will need to be merged.

4. Data Preparation

To allow ingestion into the Trimble eCognition Developer software, the required datasets (**Section 3**) should be prepared following the instructions below, and in the specified order. Note that the CRS of all datasets should be set to OSGB36/BNG (EPSG: 27700).

4.1. Root Directory

All prepared datasets should be placed in the same directory – this should not be the same directory that hosts raw, unprepared datasets ("DataDir"). This directory will be referred to as the "root directory" (or "RootDir"). Whilst the location and name of the root directory can be chosen by the user within their data management system, the input datasets must follow a strict naming convention, format, and relative folder structure within "RootDir" – these relative file paths are outlined in this section.

4.2. Vector Datasets

Vector datasets downloaded at national scale do not require clipping due to the way in which eCognition handles project creation. **Table 2** summarises the vector datasets[3] (and their layers if applicable) that should be extracted as ESRI Shapefiles and placed in the relevant sub-folder of the root directory.

Dataset	Layer Name	File Paths
BI Network	TidalWater[4]	{RootDir}\\GI\\BI_tidal_water.shp
OS Open BUA	OS_Open_Built_Up_Areas	{RootDir}\\OS\\BUA.shp
OS BNG	1km_grid	{RootDir}\\OS\\OS_grid_1km.shp
OS BNG	5km_grid	{RootDir}\\OS\\OS_grid_5km.shp
NFI	-	{RootDir}\\NFI\\NFI.shp
PHI	-	{RootDir}\\PHI\\PHI.shp
Moorland Line	-	{RootDir}\\Moorland_Line\\Moorland_Line.shp

Table 2. Summary of the input vector data (excluding GI and OSMM) required for the analysis workflow, including their relative file paths in the root directory.

4.3. Date Block Naming Convention

Larger datasets split into date blocks, such as the GI database, APGB, OSMM and EA LiDAR, must follow this naming convention to be ingested into eCognition:

{aoi}_{date}_{dataset}.{ext}

The *aoi* parameter corresponds to the name of the AOI (usually a Local Authority or Metropolitan County name). If it contains multiple words, spaces should be suppressed and upper-case letters should be used to signify the start of a word, e.g., GreaterManchester or WestMidlands.

[3] Excludes GI and OSMM which are too large.
[4] Note that preparing TidalWater is only required for coastal AOIs.

The *date* parameter corresponds to the APGB date block (**Section 3.4**) and must be in the form yyyymmdd, e.g., 20200513. There should be a file for each date present in the APGB imagery.

The *dataset* parameter corresponds to the dataset name, which can contain multiple words separated by an underscore, and the *ext* parameter corresponds to the dataset format extension. These are dataset specific and will be specified in this section.

For example, if Greater Manchester has three APGB date blocks (e.g., 20190623, 20200516, 20210421), there will be three different filenames for the APGB CIR virtual mosaics:

- GreaterManchester_20190623_APGB_CIR.vrt
- GreaterManchester_20200516_APGB_CIR.vrt
- GreaterManchester_20210421_APGB_CIR.vrt

4.4. APGB

The APGB data preparation is automated using the Python script *prepare_apgb_by_date.py* (**Appendix A**). All 1km tiles belonging to the same date block are merged into a virtual mosaic (VRT), and this is repeated for both RGB and CIR datasets. It can be extended to the DTM and DSM datasets if required (**Section 3.4**). The relative APGB file paths are summarised in **Table 3**.

Dataset	Required?	File Paths
APGB RGB	Yes	{RootDir}\\APGB\\RGB\\{aoi}_{date}_APGB_RGB.vrt
APGB CIR	Yes	{RootDir}\\APGB\\CIR\\{aoi}_{date}_APGB_CIR.vrt
APGB DTM	No - optional	{RootDir}\\APGB\\DTM\\{aoi}_{date}_APGB_DTM.vrt
APGB DSM	No - optional	{RootDir}\\APGB\\DSM\\{aoi}_{date}_APGB_DSM.vrt

Table 3. Summary of the input APGB datasets for the analysis workflow, including their relative file paths in the root directory (both required and optional are included).

The APGB dataset must be prepared first as the python script generates a list of 1km OS grid tiles corresponding to each date block. These lists are required to prepare the remaining datasets in **Section 4**.

4.5. AOI

The AOI is user defined. It should have been prepared in **Section 3.2** and its 500m buffer should have been used to download the APGB, GI, OSMM and EA LiDAR datasets.

The first two rows of **Table 4** summarise the relative file paths for the AOI and its 500m buffer. These two files are the only ones that do not follow the standard naming convention, as they have not been split by APGB date blocks and do not contain the *date* parameter.

For ingestion into eCognition, the existing 500m buffered AOI will need to be split up between the different APGB date blocks. This is automated using the Python script *create_apgb_date_blocks.py* (**Appendix B**) and requires:

- The 1km OS grid tiles to be downloaded and placed in the correct location (**Section 4.2**).
- The APGB imagery to be prepared (**Section 4.4**).
- The full 500m buffered AOI to be placed in the correct location (**Table 4**).

The script should create a new AOI file for each date blocks present in the APGB imagery, following the naming convention in the last row of **Table 4**. Note that if the AOI only contains a single APGB image date, the 500m buffered AOI will not be split up, but a copy of the existing file will be created with the *date* parameter.

Dataset	File Paths
Full AOI	{RootDir}\\AOI\\{aoi}_AOI.shp
Full AOI buffered to 500m	{RootDir}\\AOI\\{aoi}_AOI_500m_buffer.shp
500m buffered AOI split by date block	{RootDir}\\AOI\\{aoi}_{date}_AOI_500m_buffer.shp

Table 4. Summary of the AOI files and their relative file paths in the root directory.

4.6. GI

The Map1_OGL layer should be extracted from the GI Access maps database as an ESRI Shapefile and clipped to each APGB date block. At present, this is done manually using a GIS software. **Table 5** summarises the relative file path. Note that converting from a geodatabase to a Shapefile will cut-off any attribute field names longer than 12 characters, but these will be restored during the final output preparation (**Section 8.3**).

Dataset	Layer Name	File Paths
GI Access maps	Map1_OGL	{RootDir}\\GI\\{aoi}_{date}_GI.shp

Table 5. Summary of the input GI data for the analysis workflow, including its relative file path in the root directory.

4.7. OSMM

The osmm_area layer should also be extracted from the OSMM dataset as an ESRI Shapefile and clipped to each APGB date block. **Table 6** summarises the relative file path.

Dataset	Layer Name	File Paths
OSMM	osmm_area	{RootDir}\\OS\\OSMM\\{aoi}_{date}_OSMM.shp

Table 6. Summary of the input OSMM data for the analysis workflow, including its relative file path in the root directory.

4.8. EA LiDAR

Preparation of the EA LiDAR data is also automated using the Python script *prepare_ea_lidar.py* (**Appendix C**). All 5km tiles belonging to the same date block are unzipped and merged. This is repeated for both the DTM and point cloud, generating GeoTIFF files and LAZ files respectively. The relative file paths are summarised in **Table 7**.

Dataset	File Paths
DTM	{RootDir}\\EA_LiDAR\\DTM\\{aoi}_{date}_EA_DTM.tif
Point cloud	{RootDir}\\EA_LiDAR\\LAZ\\{aoi}_{date}_EA_LAZ.laz

Table 7. Summary of the input EA LiDAR datasets for the analysis workflow, including their relative file paths in the root directory.

5. Grassland Desk-Based Survey

The purpose of this survey is to collect sample polygons for the two detailed Grassland classes, Amenities (A1) and Undifferentiated (A2). We recommend carrying it out in QGIS since the proposed method requires QGIS styling. All the necessary survey files can be found in the Grassland Desk-Based Survey subfolder of the QGIS folder. One survey should be performed for each date block. It is also important to collect an equal spread of samples for each class.

5.1. Survey Set-up

5.1.1. Saving Samples

Users should first make a copy of the **Example_Shapefile_Template - Grassland_Survey.shp** template for each date block within the AOI and place these files in "RootDir" with the following naming convention:

{RootDir}\Samples\{aoi}_{date}_samples.shp

The number of samples shapefiles should match the number of date blocks.

5.1.2. Survey Files

A blank QGIS project should be created for each date block within the AOI and the required files should be loaded:

- **Samples shapefile** – This relates to the template saved in **Section 5.1.1**, which is designed to be populated with the grassland sample polygons. The correct samples shapefile for the date block should be added to the project (e.g., via drag and drop).
- **Survey styling (Grassland_Samples_Styling.qml)** – This style needs to be applied to the survey shapefile template. It will not work on other shapefiles; they need to have the required attributes.
- **Broad and Detailed Value Maps (Broad_value_map_grasslands.csv** and **Detailed_value_map_grasslands.csv)** – These data tables **must** be included as layers in the survey QGIS project. They store class keys and names, which are necessary for the styling to work properly.

The recommended survey template, styling and its associated CSV data tables allow for easier sample polygon classification. If set up correctly, drawing a sample polygon will pop-up a window where its class can be assigned from a series of drop-down options, instead of typing them manually (**Figure 3**).

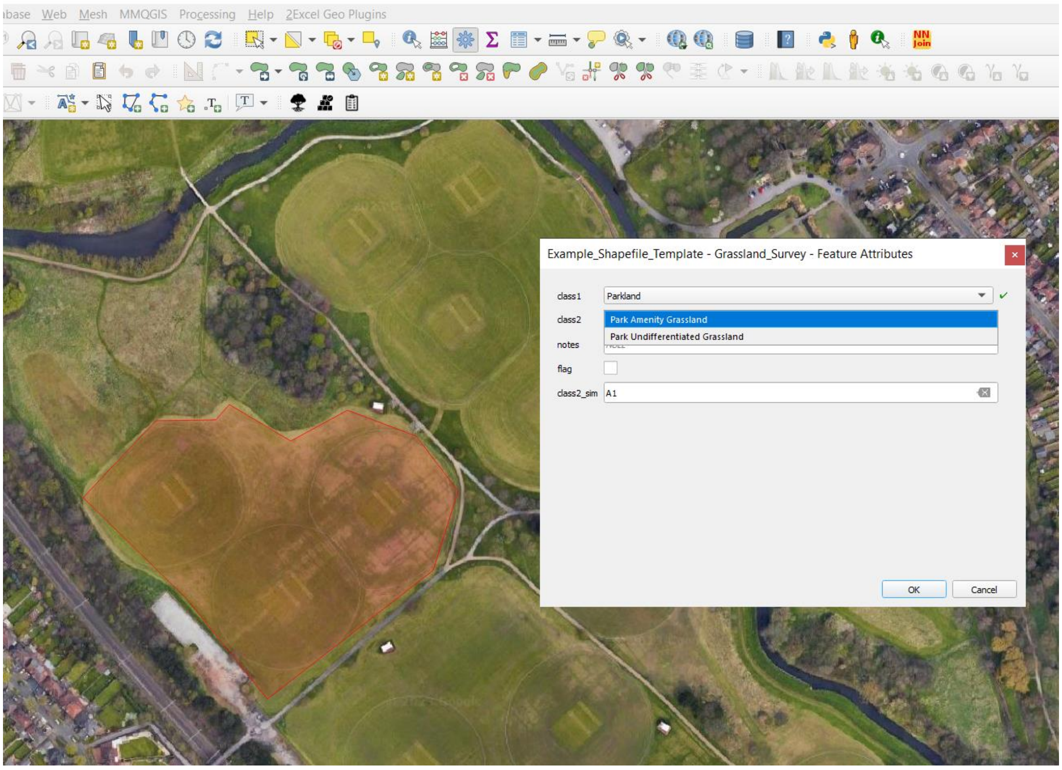


Figure 3. The drop-down options available for classifying newly drawn grassland sample polygons, if the Grassland_Samples_Styling.qml is correctly applied to the Survey shapefile template.

5.1.3. Supporting Files

A series of additional supporting files should also be loaded into the QGIS project. These can be found in the "RootDir":

- **APGB RGB Imagery** – {RootDir}\\APGB\\RGB\\{aoi}_{date}_APGB_RGB.vrt – this serves as a base map.
- **Date Block AOI** – {RootDir}\\AOI\\{aoi}_{date}_AOI_500m_buffer.shp – this constrains the area for analysis and users should not draw samples outside of it.
- **GI Access Map** – {RootDir}\\GI\\{aoi}_{date}_GI.shp – this provides contextual information in relation to the existing GI database and allows a user to decide if a grassland polygon sits with a park.
    - The "**GI_Classes.qml**" styling should be applied to the GI. All the polygons outlined in green show Parklands.

An example of how the layer tree in the QGIS project should look with all the files is shown in **Figure 4**. Users should make sure they have loaded the correct files for the date block that they are working on.

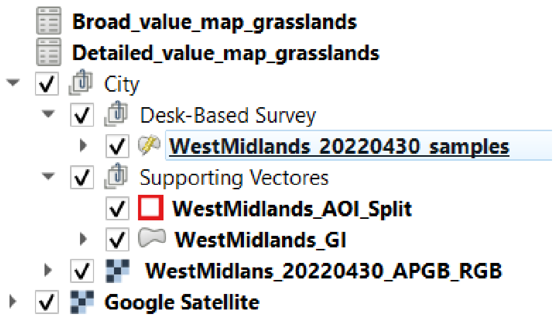


Figure 4. Example QGIS project layer layout with all the required files for the grassland desk-based survey.

5.2. Collecting Samples

To start collecting samples, edits on the survey template should be turned on. Polygons can then be digitised by visually following grassland boundaries. The size of the polygons does not matter as they will later be re-segmented in eCognition, and the GI parcel boundaries should be ignored as they may not always be representative of grassland types. Users should aim to digitise a minimum of 50 polygons for each grassland class (amenities and undifferentiated) roughly spread out over the entire date block, regardless of context (within vs outside of parks). However, this number is dependent on the size of the date block, the abundance of grasslands and the respective sizes of these grassland habitats. Users should ensure they save their edits regularly to avoid any loss of work.

6. Analysis Workflow

6.1. eCognition Basics

The following section relies on the use of the Trimble eCognition Developer and Server software. It is intended to be used by someone who possesses basic familiarity with the software. To gain or expand one's familiarity, we recommend using Trimble's User Guide and video tutorials:

- Trimble eCognition Help - eCognition Suite Overview
- https://www.youtube.com/@eCognitiontv/featured

As mentioned in **Section 2.4**, there are two ways to analyse projects in eCognition, either locally in the Developer, or by submitting projects to the Server:

Developer Analysis

To analyse via eCognition Developer, a project needs to be opened within a workspace. The ruleset of interest should then be loaded in the Process Tree panel (**Figure 5**). This view enables the user to execute any parts of the ruleset in isolation. To run the full ruleset rather than parts of it, a user should execute the process from the very top.

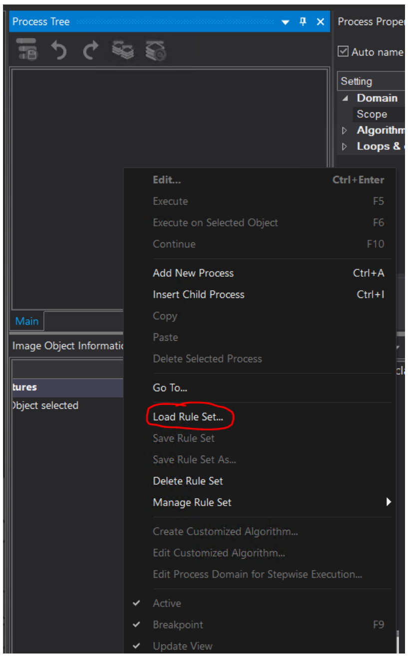


Figure 5. Any rulesets can be loaded into the Process Tree within an eCognition workspace.

Server Analysis

Within an eCognition workspace, multiple projects can be selected at once and submitted for analysis (**Figure 6**).

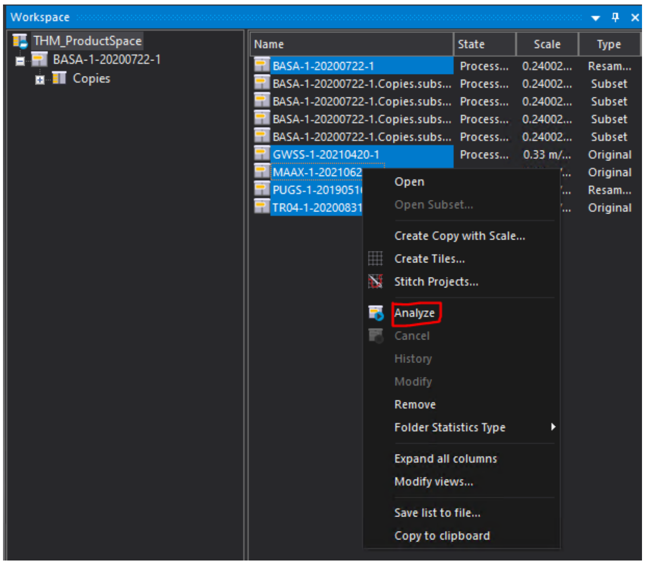


Figure 6. Multiple projects have been selected for analysis using eCognition Server.

The **Start Analysis Job** window that pops-up (**Figure 7**) allows the GRID server network to be specified (it is set to local server by default, meaning that the jobs are submitted to the user's machine rather than a network). The file path to a ruleset must also be specified. The chosen projects will then be analysed with the given ruleset.

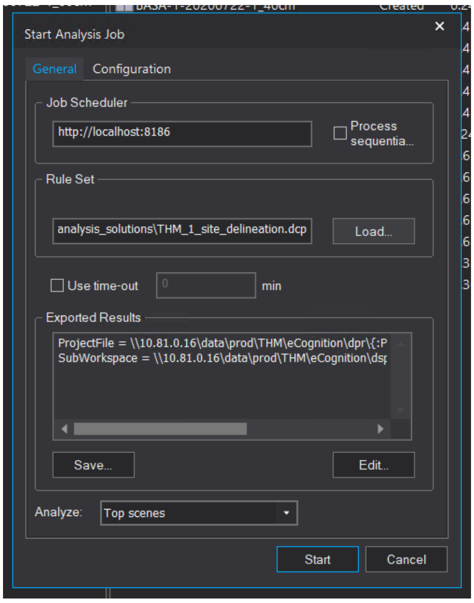


Figure 7. The "Start Analysis Job" window to send analysis jobs to server for a given ruleset.

We recommend using eCognition Developer to test individual tiles with specific parameters. Once the parameters have been optimised, it is advised to run the analysis using the eCognition Server option. As explained in **Section 2.4**, submitting rulesets to eCognition Server requires Server licenses. One license is required per "engine", so purchasing multiple Server licenses means that multiple tiles can be analysed in parallel, and processing can be left running overnight.

Note that to modify ruleset parameters, users must load the ruleset in eCognition Developer prior to submitting it to Server. The parameters should be manually modified, and the ruleset saved – this can be done in the workspace without opening a project.

6.2. Workspace and Project Set-Up

6.2.1. Creating a New Workspace

The initial step when using eCognition is to create a new "Workspace". The workspace will house all the projects, and it creates a directory where all the output and intermediate files can be found. It can be saved anywhere; it does not need to be in the same place as the input data.

6.2.2. Customised Import

Within the new eCognition workspace, the "**Customised Import**" tool needs to be used to create individual projects for each given APGB date block. **Figure 8** shows an example of the tool.

- The "**GI_import.xml**" file needs to be loaded into the tool.
- The "**Root Folder**" must specify the location of the "RootDir" where all the input datasets are saved.
- The "**Master File**" must be one of the APGB CIR VRT files (you can pick any AOI and date block). Choosing the CIR is important because the master file will dictate the spatial resolution of the project. In this case, it will be set to 50cm/pixel, and all other datasets will be resampled to that value. The master file will also dictate the import's naming convention for the remainder of the files. Note that you may get a dialog window as shown in **Figure 9** when selecting the master file. Users should press "No", otherwise the customised import will auto-reset.
- The search string may then be tested using "Test" to see if the correct file was found.
- By pressing "OK", projects will be created for every single date block that has been prepared.

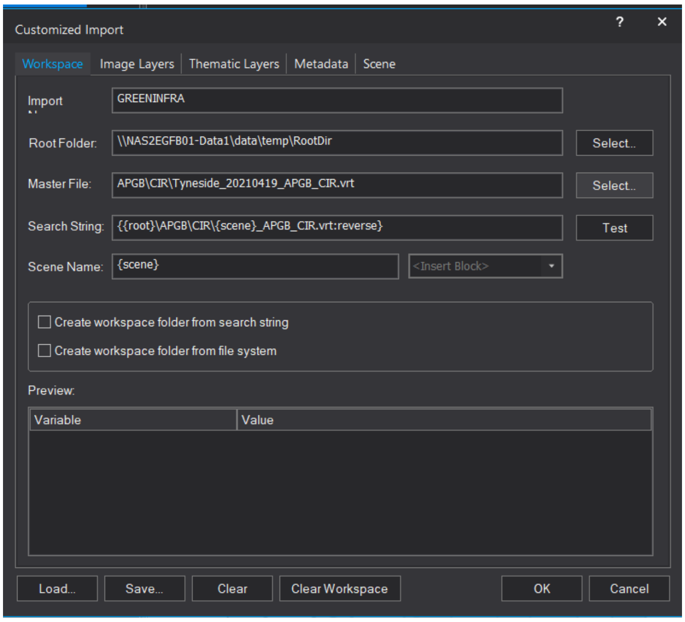


Figure 8. Example of the customised import tool.

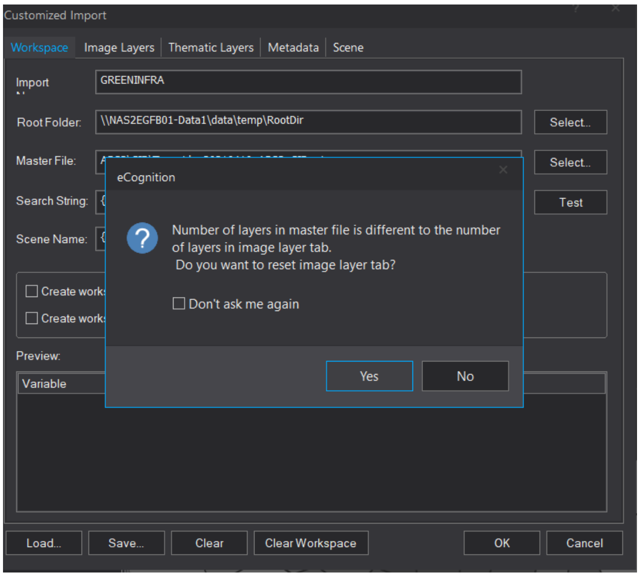


Figure 9. Example of the dialog window that may appear upon master file selection. Users should press "No".

Please see Trimble's user guide Trimble eCognition Help - Automating Data Analysis for more details on using the tool.

6.3. Grassland Statistics Extraction

Prior to classification, grassland statistics need to be created and extracted for each project (date block).

This can be done by submitting a project to Server and processing it with the "**GI_0_grass_stats.dcp**" ruleset (**Figure 10**) as explained in **Section 6.1**[5]. One parameter needs to be specified in the ruleset before analysis:

- **attribute_name** – The name of the attribute (or column) with the detailed grassland classes. The attribute name is set to **"class2_sim"** by default. If the grassland samples were collected according to **Section 5**, it can be left as default.

Once submitted, the ruleset will prepare all the data required for analysis within the project, e.g., Vegetation Indices (VI), DSM creation from the LiDAR point cloud, building and private garden extraction from the OSMM dataset. It will then perform multi-resolution segmentation, create sample objects from the **grassland samples shapefile** and update all the needed statistics of each sample object. The statistics include spectral, physical, and textural attributes and are exported as a CSV file with the naming convention **{city}_{date}_grass_samples.csv**. The file can be found in the "**samples**" folder of the eCognition workspace directory.

[5] Alternatively, this can be done in eCognition Developer, but opening a date block project may take some time.

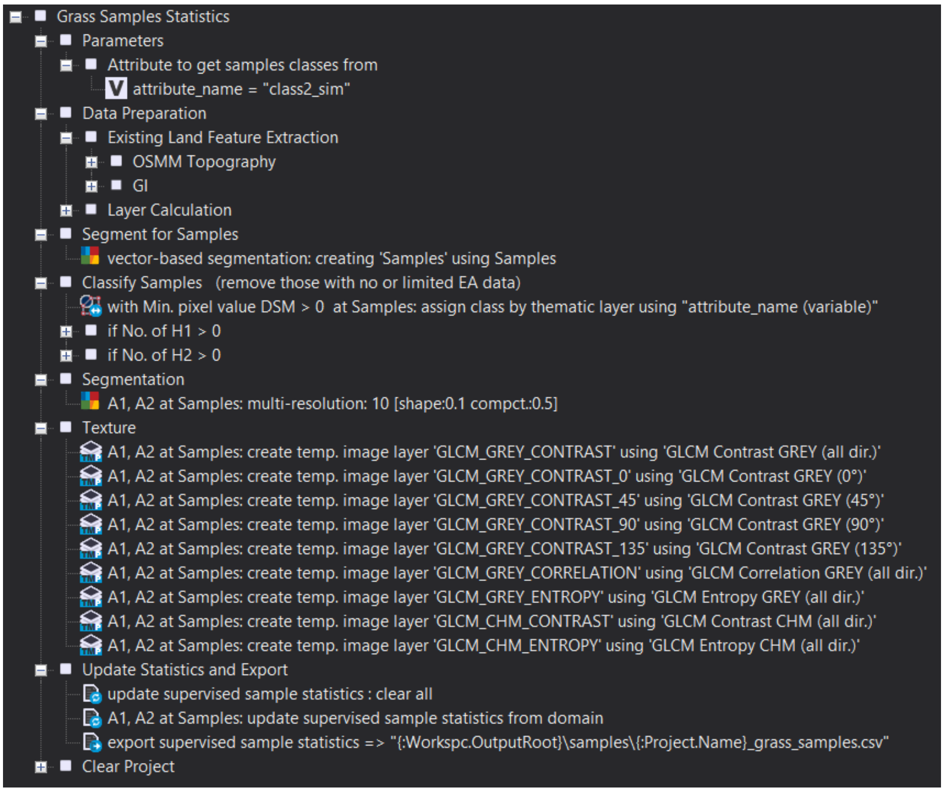


Figure 10. The "GI_0_grass_stats.dcp" ruleset used for extracting grassland statistics.

6.4. Tiling

Once the grassland statistics have been derived on a date block level, the projects need to be split-up into individual 5km OS grid tiles for further analysis. This will create a series of sub-projects, each named according to their given OS tile:

{city}_{date).Copies.subset.{OS tile}

Projects are tiled by submitting them to Server and processing them with the "**GI_1_tiling.dcp**" ruleset (**Figure 11**). Note that tiling cannot be done within eCognition Developer so submission to Server here is critical (as well as the purchase of an eCognition Server license). Two parameters need to be specified prior to running the ruleset:

- **OS_5km_tile_code** – The name of the attribute (or column) with the OS grid tile code. It is set to "PLAN_NO" by default.
- **tile_buffer_m** – Tile buffering distance in meters. It is set to 500m by default.

The ruleset will then segment the project according to the geometry of the OS tiles and create new sub-projects which will appear in the workspace layout (**Figure 12**).

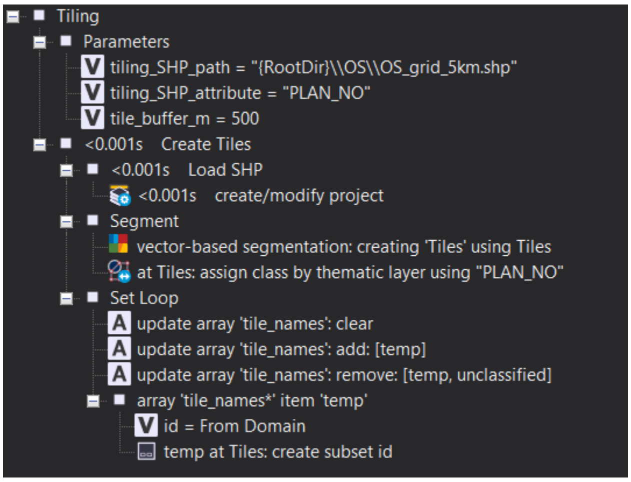


Figure 11. The "GI_1_tiling.dcp" ruleset used to split-up projects into 5km OS grid tiles.

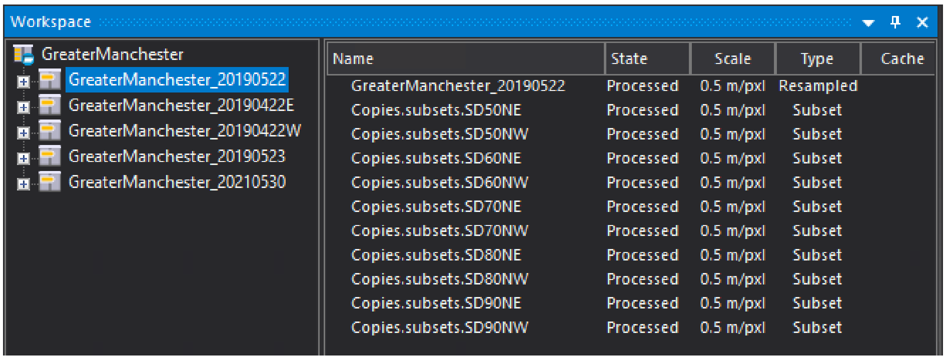


Figure 12. Example of a workspace layout for Greater Manchester, which contains 5 date blocks. Clicking on a date block project shows the sub-projects as "Copies.subsets.{OS tile}" on the right-hand side.

6.5. Urban Habitat Classification and Naturalness Mapping

The "**GI_2_classification.dcp"** ruleset carries out the actual urban habitat classification and is far more complex (**Figure 13**). It consists of multiple sections and requires the users to input parameters. Parameters are specifically tailored to each APGB date block, due to the variation in spectral characteristics across the range of image dates. The ruleset is tailored by adjusting the default spectral thresholds. This must be done manually and can only be achieved through a series of trial and error to see which threshold produces the best results. Users are advised to start off with the default parameter values and tweak accordingly. Ruleset adjustments should be performed on a subset of a project to make processing faster. **Section 6.5.2** provides detailed explanation on threshold adjustment.

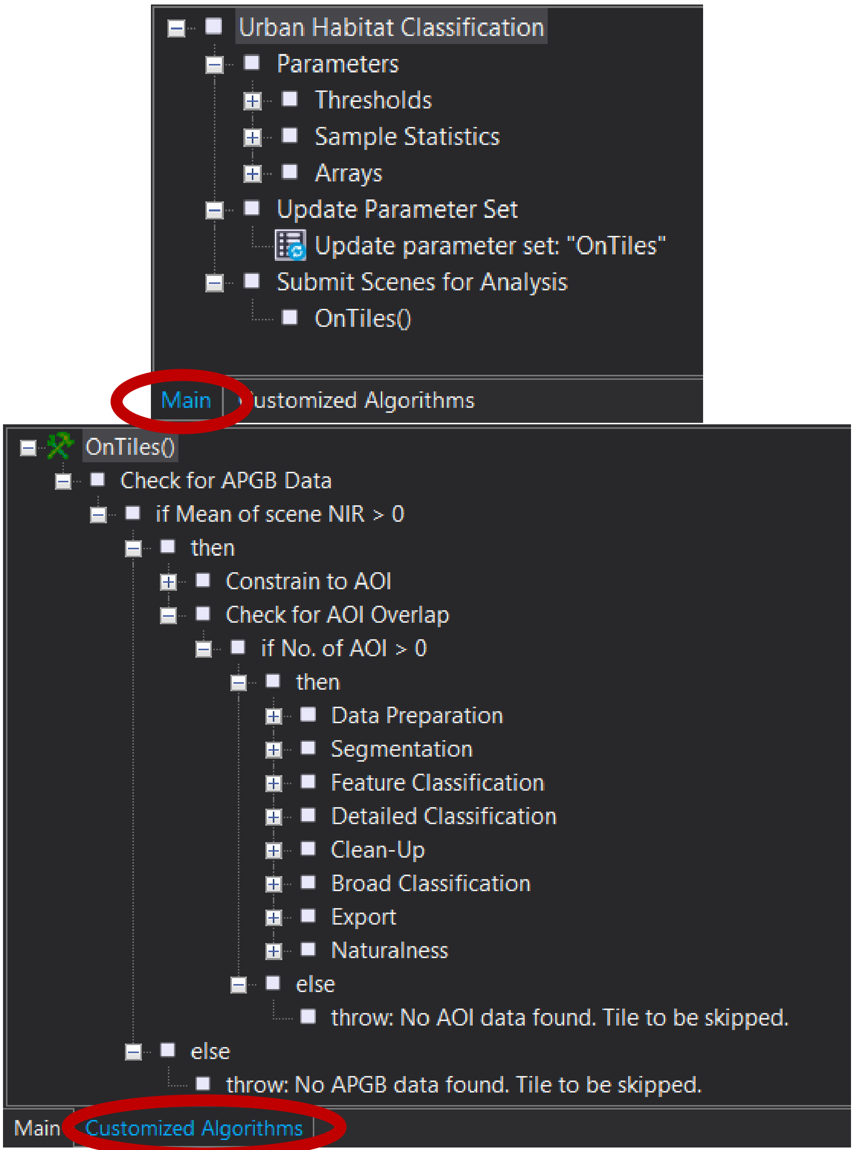


Figure 13. The "GI_2_classification.dcp" ruleset used for urban habitat classification. The section in the "Main" tap (above) contains all the parameters which require manual input. The section in the "Customised Algorithms" tab holds processes which carry out the classification and exports.

6.5.1. Creating a Subset

A subset can be created by left clicking on any project in the eCognition workspace layout and selecting "**Open Subset…**". A Subset Selection window will pop up, which allows the user to draw a subset on the image by clicking and dragging their mouse (**Figure 14**). It is best to select a subset area that has an urban area with many sealed surfaces, but also green spaces and some bare ground. Clicking "**OK**" will create the subset. Once the subset is open, it must be renamed by selecting "**File**" > "**Modify Open Project…**" and typing a Project Name. Please note that keeping the subset name as "New Project" will cause serious issues. Lastly, the "**GI_2_classification.dcp"** ruleset needs to be loaded into the Process Tree panel as shown in **Section 6.1**.

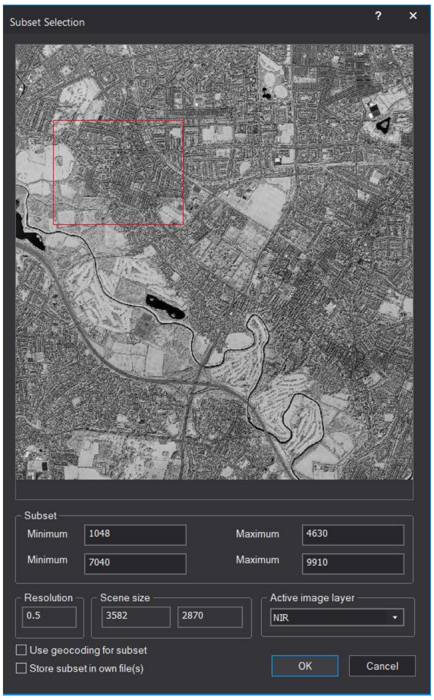


Figure 14. Subset selection window for manually drawing a subset area to create a subset project.

6.5.2. Ruleset Parameters Adjustment

The **Parameters** section of the ruleset, in the "Main" tab of the Process Tree (**Figure 11**), contains all the parameters which require manual adjustment prior to analysing any new date block.

<u>Thresholds</u>

This section of the ruleset contains all the spectral threshold variables which need to be reviewed and modified (**Figure 15**). They are separated into two types according to the ruleset sections they are used in, either the Feature or Detailed classification. The names of the threshold variables always state which spectral indices they concern. There is also a comment next to each one which specifies in which line that specific threshold is used in the ruleset. **Table 8** provides information about the different thresholds, their default values, and recommendations for optimisation**.** All parameters must have a value, and leaving one empty will throw an error.

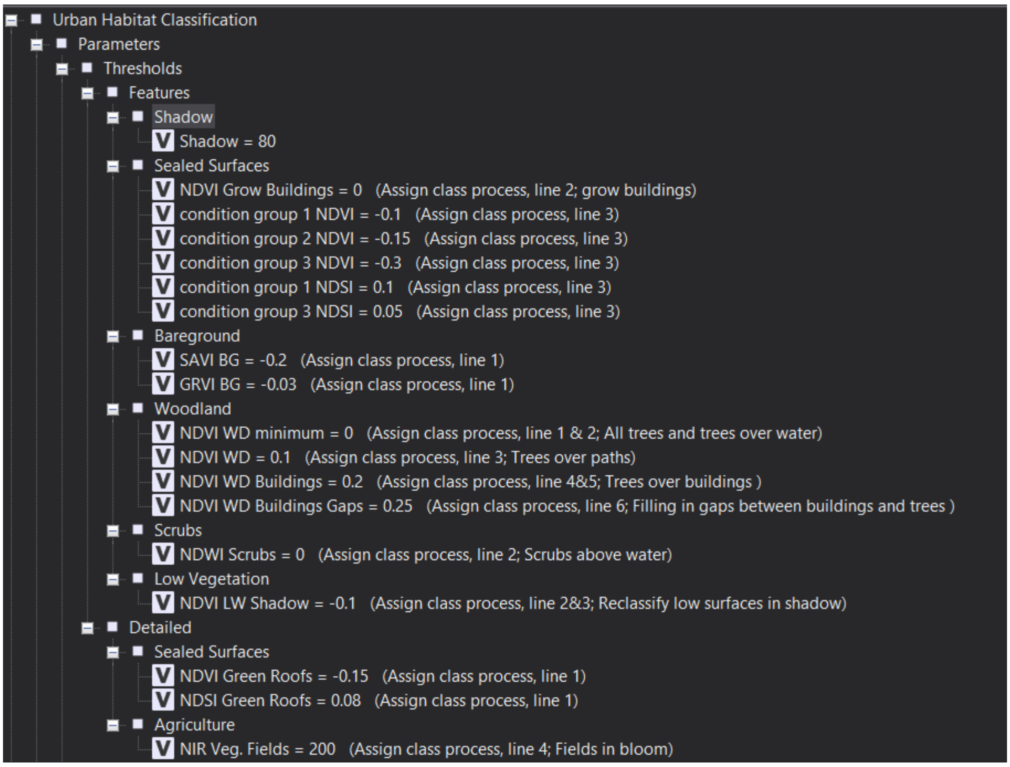


Figure 15. Threshold parameters in the “GI_2_classification.dcp” ruleset.

Classification	Parameter	Default Value	Values should be increased if:	Values should be decreased if:
Shadow	Shadow	85	Too few objects are being classified as shadows.	Too many objects are being classified as shadows.
Sealed Surfaces	NDVI Grow Buildings	0.2	All buildings remain the same as OSMM footprints.	After executing this process, trees or other green features are being misclassified as buildings.
	condition group 1 NDVI[6]	0.15	Sealed surfaces objects are not being classified as sealed surfaces.	Bare ground or low vegetation objects are misclassified as sealed surfaces.
	condition group 2 NDVI[7]	0		
	condition group 3 NDVI[8]	-0.1		
	condition group 1 NDSI	0.1	**Sealed surfaces with mean NDVI values below the condition group NDVI thresholds are still not classified as sealed surfaces.**	
	condition group 3 NDSI	0		
Bare ground	SAVI BG	0.2	Bare ground objects are not being classified as bare ground.	Low vegetation objects are misclassified as bare ground.
	GRVI BG	0		
Woodland	NDVI WD minimum	0.1	Buildings, shadows, or other objects are misclassified as woodland.	Woodland objects are not being classified as woodland.
	NDVI WD	0.3		
	NDVI WD Buildings	0.2		
	NDVI WD Buildings Gaps	0.25		
Scrubs	NDWI Scrubs	-0.1	Scrub objects above water are not being classified as scrubs.	Water objects are misclassified as scrubs.
Low Vegetation	NDVI LW Shadow	0.4	Sealed surfaces objects that are shaded are being misclassified as low vegetation.	Low vegetation objects that are shaded are not being re-classified as low vegetation.

[6] The first condition group is intended for sealed surfaces with a higher NDVI than average, which are then constrained by their NDSI value.
[7] The second condition group is meant for average NDVI sealed surfaces, and this value should therefore be lower than condition group 1.
[8] Condition group 3 is for sealed surfaces in natural spaces (e.g., parks). This value should be the lowest to reduce misclassification with bare ground and the unnecessary removal of poor/thin vegetation.

Green Roofs	NDVI Green Roofs	0.15	Building objects are being misclassified as green roofs.	Green roof objects are not being classified as green roofs.
	NDSI Green Roofs	-0.01		
Vegetated Fields	NIR Veg. Fields	200		Fields in bloom are not being re-classified as vegetated fields and remain as ploughed fields.
Coastal	NDVI Coastal LW	0.1		Low vegetation in coastal areas is not being re-classified as low vegetation.
	NDVI Coastal Shadows LW	0.35		
	NDSI Sand	0.08	-	-

Table 8. List of parameters, their default values, and recommendations for optimisation.

When adjusting thresholds, a subset project should be opened and the "**GI_2_classification.dcp**" ruleset should be loaded in as shown in **Section 6.1**. The default parameter values should be used to start with. The ruleset should then be executed up to the "**Feature Classification**" section in the Customized Algorithm tab. This can be done by adding a "**Breakpoint**" to the ruleset at the "**Feature Classification**" step, then executing the ruleset as normal from the top (**Figure 16**).

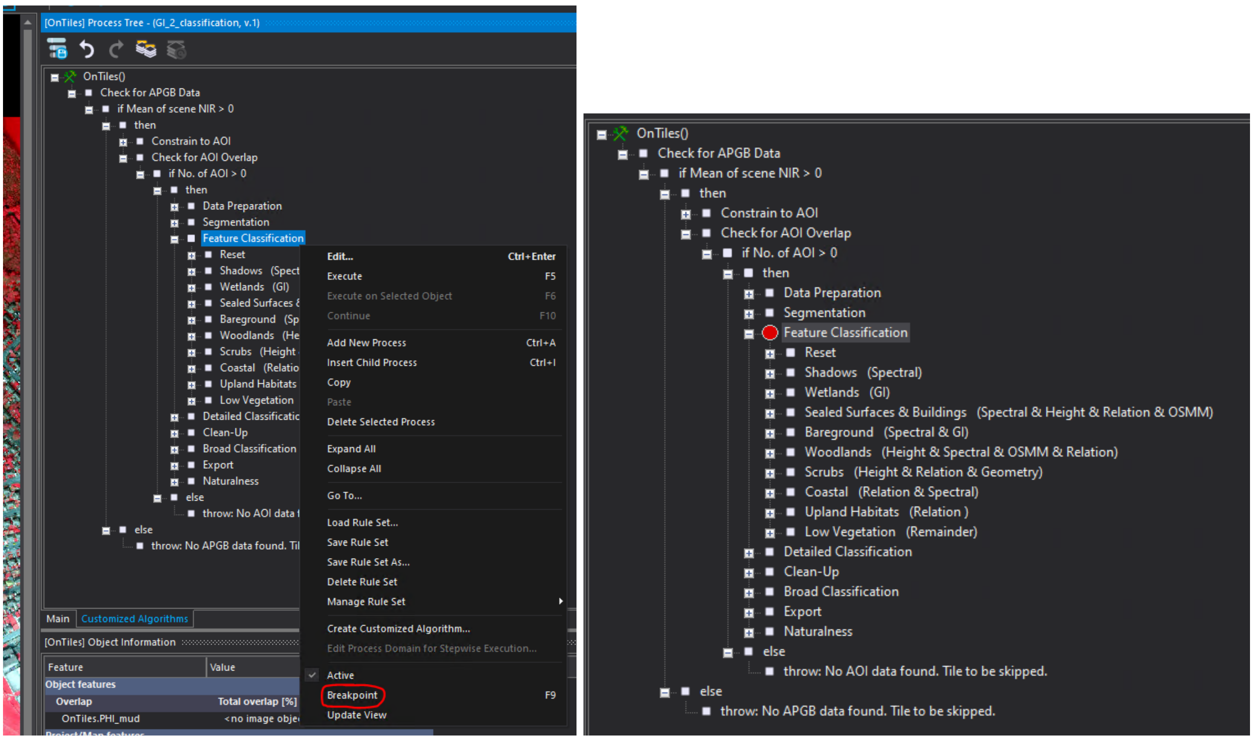


Figure 16. Adding a breakpoint in the ruleset (left) results in a red circle next to the chosen stop point (right). Users can then execute all processes up to that point by running the ruleset from the top.

Once the start of the ruleset has completed, users should see two object levels: AOI and Features. As shown in **Figure 17**, the AOI object level should be unchecked, and the Features object level should be checked in the View Settings panel. The outline should also be unchecked. A solid white fill should be seen across the image.

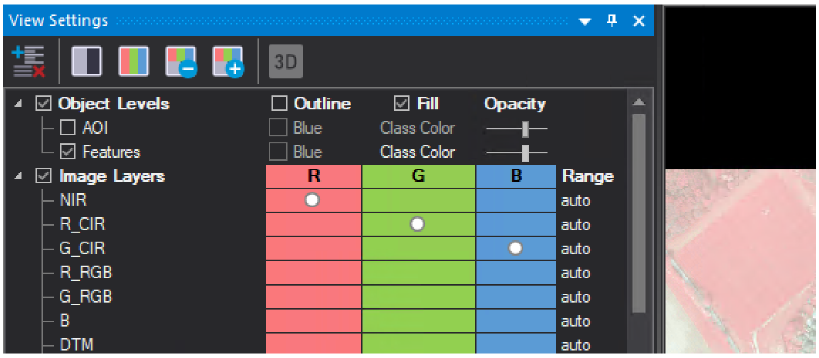


Figure 17. After executing the start of the ruleset up to the breakpoint, two object levels should be visible.

Using the **"Shadow"** variable as an example here, the "**Shadow (Spectral)**" process in "**Feature Classification**" can then be executed individually. This will classify all objects in the **Features** object level with a brightness value below the "**Shadow**" threshold to be classified as "**Shadows**". The shadow objects should become visible in black (**Figure 18**).

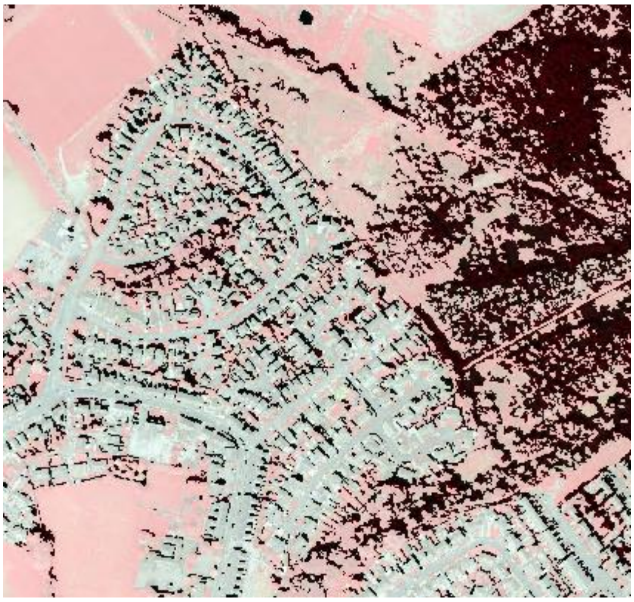


Figure 18. Example of what shadow objects would look like after executing the "Shadow (Spectral)" process.

Users should assess the shadows visually. If satisfactory, the default parameter can be used. Otherwise, the threshold should be modified. To do that, the value of the parameter in the "Main" tab should be updated (**Figure 19**) – make sure to press "Execute", otherwise the new threshold value will not be saved.

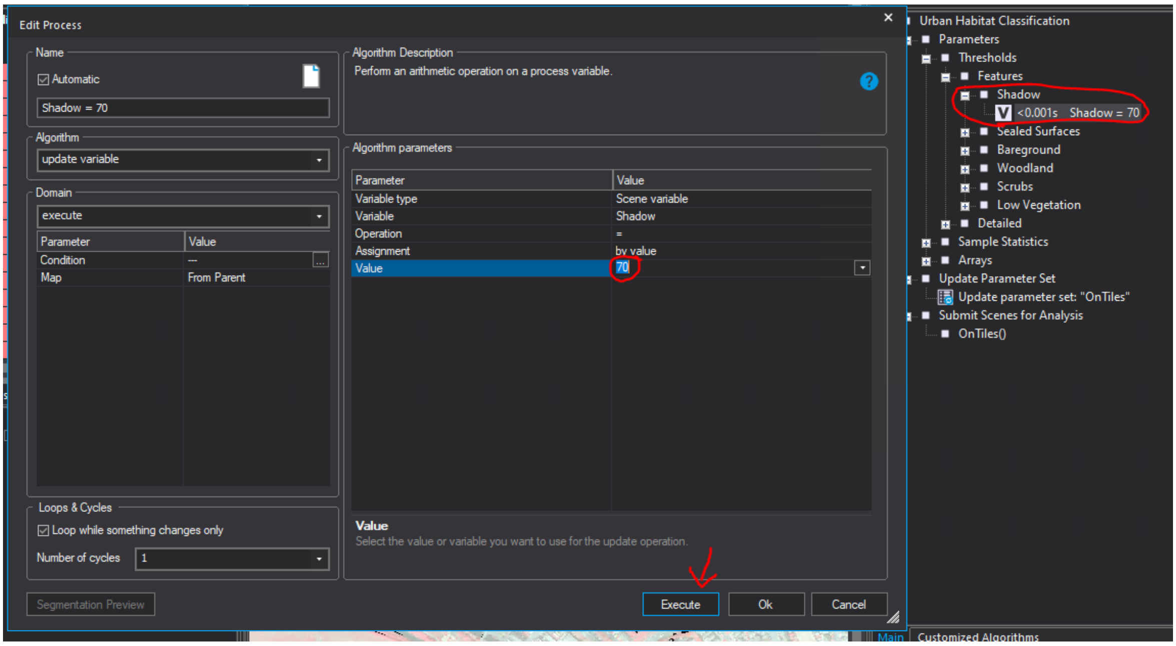


Figure 19. Example of updating the shadow parameter threshold value. Execute must be pressed to save the change before re-classifying.

Users should then move back to the “Customized Algorithm” tab and execute the “**Reset**” process in “**Feature Classification**” (**Figure 20**), followed by the “**Shadow (Spectral)**” process again. This process should be repeated until users are satisfied with the shadow classification.

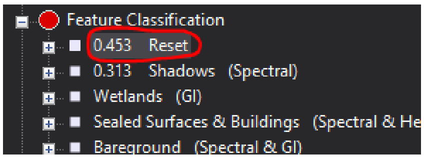


Figure 20. Executing “Reset” will reset the Features classification so that the shadows can be re-classified with a new threshold.

Please note that shadows are classified based on their spectral brightness. It is recommended that the “BRIGHTNESS” image layer is viewed during the modification to help inform decisions. **Figure 21** shows how the View Settings panel should look like. If the black shadow objects are difficult to see, their colour can be changed.

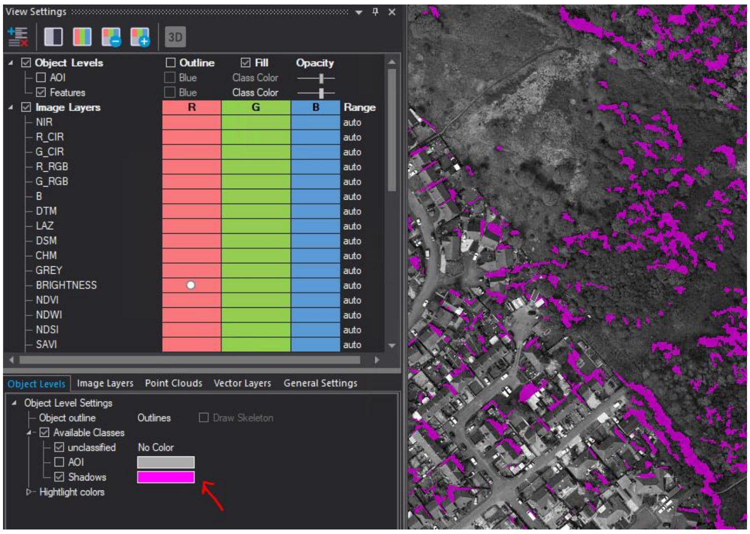


Figure 21. State of the View Settings panel to visualise shadows from the Features object level in magenta, with the "BRIGHTNESS" layer as underlay.

Additionally, the mean "BRIGHTNESS" value can be displayed in the Object Information panel for a selected object. Simply left click in the panel, choose "**Select Features to Display…**" and navigate to mean BRIGHTNESS in the **Object Features > Image Layer** (**Figure 22**). Users might want to turn the outline of the Features objects back on in the View Settings panel. It makes selecting objects easier.

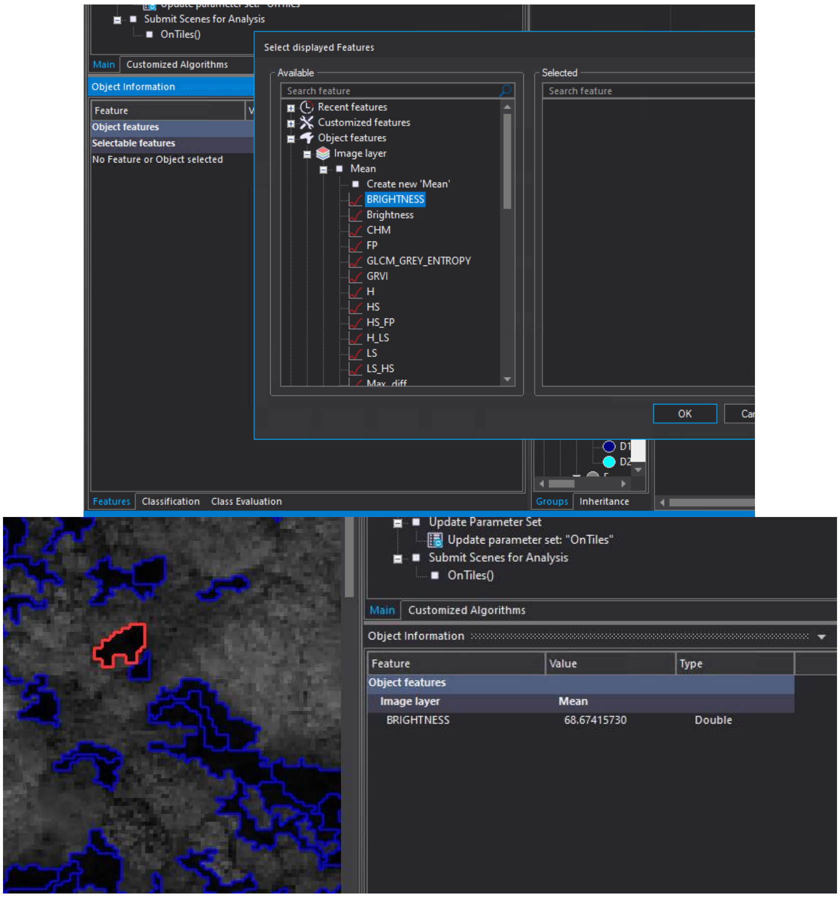


Figure 22. Adding mean BRIGHTNESS in the Object Information panel and selecting shadow objects to view the brightness values.

The same process is used when adjusting thresholds for the other parameters. The only difference is that all features prior to the feature of interest must be executed every time the classification is reset, e.g., when adjusting values for sealed surfaces, the shadows and wetlands processes must be run first in chronological order every time.

<u>Sample Statistics</u>

The one parameter in the **Sample Statistics** section specifies the file path that should be used to retrieve the grassland statistics. This needs to be updated for each date block and should point to the CSV statistics file which matches the specific date block being classified (**Figure 23**). The files are found in the "**samples**" folder inside the eCognition workspace directory as explained in **Section 6.3**.

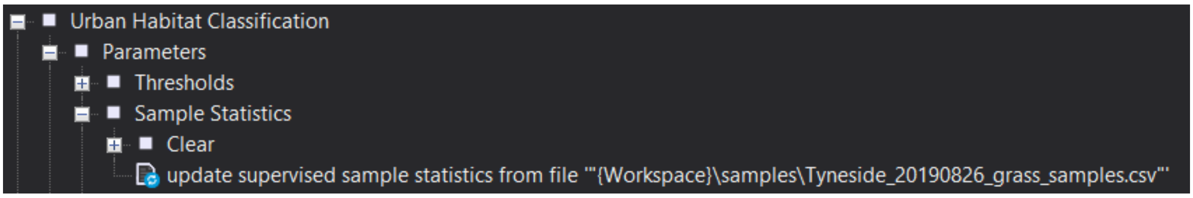


Figure 23. Sample Statistics parameter in the "GI_2_classification.dcp" ruleset.

Once all the ruleset parameters have been updated and optimised for a given date block, the ruleset should be saved. We recommend saving all rulesets within the eCognition workspace directory, in a "rulesets" folder. Users may wish to save a copy of the original ruleset in order to retain a version with the default parameter values. The newly modified ruleset can be saved with the date block name as a suffix:

GI_2_classification_{aoi}_{date}.dcp

6.5.3. Tile Analysis

Around 1-3 sub-projects within a given date block should then be submitted to Server for analysis using the modified ruleset that matches that date block. The outputs will be assessed before rolling out the ruleset to all sub-projects within the date block (**Section 7**). All the classification outputs will appear in the "**results**" folder of the eCognition workspace directory. After a tile sub-project is processed, it will be saved, and all the object levels will remain. Users can open a sub-project and view the outputs within the eCognition software.

6.5.4. Outputs

The are four main outputs produced by executing the "**GI_2_classification.dcp"** ruleset, which can all be found it the "**results**" folder of the eCognition workspace directory:

- **Urban Habitat Map – Vector File:** Both the broad and detailed classification maps are exported within a single geodatabase (GDB) file. They are exported to the "**GDB**" folder:

{Workspace}\results\GDB\{aoi}_{date}.Copies.subsets.{OS tile}_Classification.{version}.gdb

- **Urban Habitat Map – Raster File:** The broad and detailed classification maps are exported as separate raster TIF images, together with a CSV value map to specify the class associated with each pixel value in the raster. They are exported to the "**TIF**" folder:

{Workspace}\results\TIF\{aoi}_{date}.Copies.subsets.{OS tile}_Classification.{version}.tif

- **Aggregated Naturalness Map:** The aggregated naturalness map is exported as a raster TIF image, where the pixel value corresponds to the naturalness score. They are exported to the **"Naturalness"** folder:

{Workspace}\results\Naturalnesss\{aoi}_{date}.Copies.subsets.{OS tile}_Classification.{version}.tif

- **Green Infrastructure Parcels:** The percentage cover of every naturalness class (1 to 6) and the combined naturalness score (N Factor) of each GI parcel is exported. One CSV file is produced for every type of typology present in the original GI database (e.g., Local Nature Reserve). This is done to tackle overlaps between multiple typologies in the same area on the ground. One sub-project can therefore have up to 19 different CSV files produced. They are all exported to the "**GI_Parcels**" folder:

{Workspace}\results\GI_Parcels\{aoi}_{date}.Copies.subsets.{OS tile}_Classification_{typology}.csv

Note that the "version" parameter in the naming convention relates to the eCognition sub-project version. This means that when reprocessing, files are not overwritten. The only exception is with the GI parcels CSV files – these are always overwritten.

7. Accuracy Assessment

As with the Grassland Desk-Based Survey, it is recommended to carry out the accuracy assessment in QGIS. All the required styling can be found in the accuracy assessment subfolder of the QGIS folder. For each date block, 1-3 tiles should be analysed and assessed first. Based on the accuracy assessment results, the ruleset can be further modified and improved, before analysing all tiles. We recommend creating an accuracy assessment directory to store QGIS projects and assessed tiles. This does not have to be within the eCognition workspace directory.

7.1. Project Set-up

7.1.1. Survey Files

A blank QGIS project should be created for each tile to be assessed and the required files should be loaded:

- **Detailed Urban Habitat Map** – The output vector detailed urban habitat classification map from eCognition (found in eCognition workspace directory > results > GDB) should be loaded into the project (e.g., via drag and drop). To avoid accidently changing or deleting the classification, the detailed map should be re-exported as a shapefile and saved in the newly created accuracy assessment directory. The accuracy assessment should be carried out on the Shapefile copy, not the original GDB file.
- **Survey styling (Detailed_Classes_for_Assessment_Styling.qml)** – This style needs to be applied to the detailed urban habitat map layer in the project.
- **Broad and Detailed Value Maps (Broad_value_map.csv** and **Detailed_value_map(v2).csv) –** These data tables **must** be included as layers in the survey QGIS project. They store class keys and names, which are necessary for the styling and therefore the assessment to work properly.

7.1.2. Supporting Files

A series of additional supporting files should also be loaded into the QGIS project. Some of these can be found in the "RootDir" and are summarised in **Table 9**, along with the recommended style file.

File	Location	Styling
APGB RGB Imagery	{RootDir}\\APGB\\RGB\\{aoi}_{date}_APGB_RGB.vrt	
Date Block AOI	{RootDir}\\AOI\\{aoi}_{date}_AOI_500m_buffer.shp	
OS Open BUA	{RootDir}\\OS\\BUA.shp	
Moorland Line	{RootDir}\\Moorland_Line\\Moorland_Line.shp	
GI Access Map	{RootDir}\\GI\\{aoi}_{date}_GI.shp	GI_Classes.qml
NFI	{RootDir}\\NFI\\NFI.shp	NFI_Classes.qml
PHI	{RootDir}\\PHI\\PHI.shp	PHI_Classes.qml

Table 9. Files from the root directory required for the accuracy assessment.

The remainder of the supporting files can be found in the eCognition workspace directory. They are summarised in **Table 10**, along with the recommended style file.

File	Location	Styling
OSMM Buildings	{Worspace}\\intermediates\\vectors\\{aoi}_{date}_OSMM_Buildings.shp	OSMM_Buildings.qml
OSMM Gardens	{Worspace}\\intermediates\\vectors\\{aoi}_{date}_OSMM_Gardens.shp	OSMM_Gardens.qml
CHM	{Worspace}\\intermediates\\rasters\\{aoi}_{date}_CHM.tif	CHM.qml

Table 10. Files from the eCognition Workspace required for the Accuracy Assessment.

The following additional layers are highly recommended and can be freely added from QGIS using the QuickMapServices plugin:

- OSM base map
- Google Satellite base map

An example of how the layer tree in the QGIS project should look with all the files is shown in **Figure 24**. Users should make sure they have loaded the correct files for the tile that they are working on.

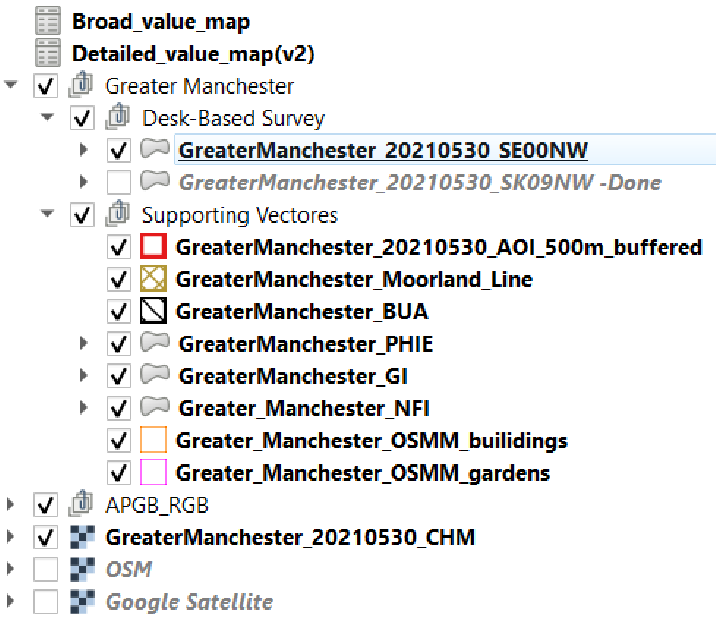


Figure 24. Example QGIS project layer layout with all the required files for accuracy assessment.

7.2. Assessment Method

Once the QGIS project is properly set up, it is recommended that 10 samples of each detailed class are manually assessed per tile. The "Predicted Broad Class" and "Predicted Detailed Class" columns in the attribute table show the classes assigned by the eCognition analysis workflow. The assessor is given a choice to either agree with the classification and tick the "Do you agree?" box or to disagree and choose the actual broad and detailed classes from the drop-down options.

Please see **Appendix G** for a detailed step by step guide on how to carry out the assessment in QGIS.

7.3. Confusion Matrix

The confusion matrix and overall accuracy can be obtained for each assessed tile using the Python script *generate_confusion_matrix.py* (**Appendix D**). An example is shown in **Figure 25**.

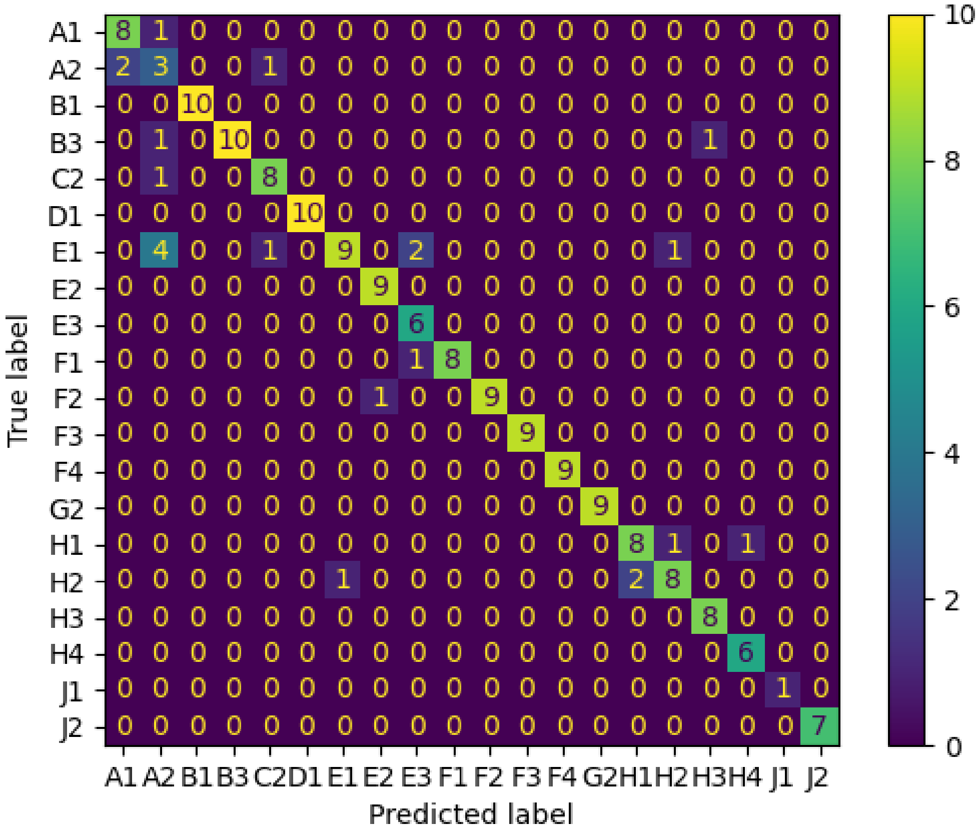


Figure 25. Example of an output confusion matrix, saved as a Portable Network Graphics (PNG) file.

If the accuracy assessment is reasonable across the tiles assessed, the ruleset and selected thresholds can be rolled-out to the remaining tiles. Otherwise, the thresholds should be modified and the tiles should be re-assessed until the outputs are satisfactory.

8. Outputs Preparation

It is recommended to manually place the outputs in a different directory, outside of the eCognition workspace, which will be referred to as the "output directory" or "OutDir". This will avoid any potential output being over-written when re-executing a ruleset.

8.1. Urban Habitat Map

The Urban Habitat Map is exported as individual 5km OS grid tiles in the geodatabase (GDB) file format. Each tile contains both broad and detailed classification levels as separate layers. The final tiles should be selected from the eCognition results directory and copied into the following directory, regardless of APGB date block:

{OutDir}\Urban Habitat Map\{aoi}

As the eCognition tile output naming convention cannot be modified within the software, the Python script *rename_gdb_tiles.py* (**Appendix E**) is used to automatically rename all tiles to a preferred naming convention:

{aoi}_{date}_{os_grid}_Urban_Habitat_Map.gdb

Tiles do not need to be merged, making it easier for delivery and dissemination.

8.2. Naturalness Map

The Naturalness Map is also exported as 5km OS grid tiles, but they should be merged into one final GeoTIFF (TIF) file, regardless of APGB date block. The final Naturalness tiles should be manually selected from the eCognition results directory and merged in a GIS software. The final TIF file should be saved in:

{OutDir}\Aggregated Naturalness Map\{aoi} - Aggregated Naturalness Map.tif

8.3. GI Parcels

The GI Parcels are also exported as 5km OS grid tiles. They should be combined into one final GeoPackage (GPKG) file for each APGB date block using the Python script *calculate_combined_naturalness.py* (**Appendix F**). The script ensures that GI parcels split between OS grid tiles are dissolved and their Naturalness factor is re-calculated. It also ensures the attribute field names that have been cut-off in **Section 4.6** match the original GI database. The GI parcel tiles do not need to be manually selected, as this is done automatically in the Python script. The output filepath by default is:

{OutDir}\Green Infrastructure Parcels\{aoi} - Green Infrastructure Parcels.gpkg

Note that the output GeoPackage will contain a "fid" index which can be deleted, and the "OBJECTID" field can be reset as the index instead.

9. Summary

At the end of this workflow, the users should end up with the following three datasets in their output directory for their chosen urban area of England:

- Urban Habitat Map (**Figure 26**) – geodatabase (GDB) files as 5km OS grid tiles containing classified urban habitat parcels at the broad and detailed levels as defined by the classification scheme (**Section 1**).
- Aggregated Naturalness Map (**Figure 27**) – GeoTIFF file with pixel values ranging from 1 to 6, indicating the level of Naturalness, where 1 is the highest Naturalness level and 6 is the lowest. This output is derived from the detailed urban habitat classification map, where each habitat class is associated with a Naturalness score.
- Green Infrastructure Parcels (**Figure 28**) – GeoPackage (GPKG) file which matches the original GI database, with the addition of 7 new fields, including the percentage cover of each Naturalness score (from 1 to 6) present within each parcel and a combined (weighted) Naturalness factor. This output is derived from the aggregated Naturalness map.

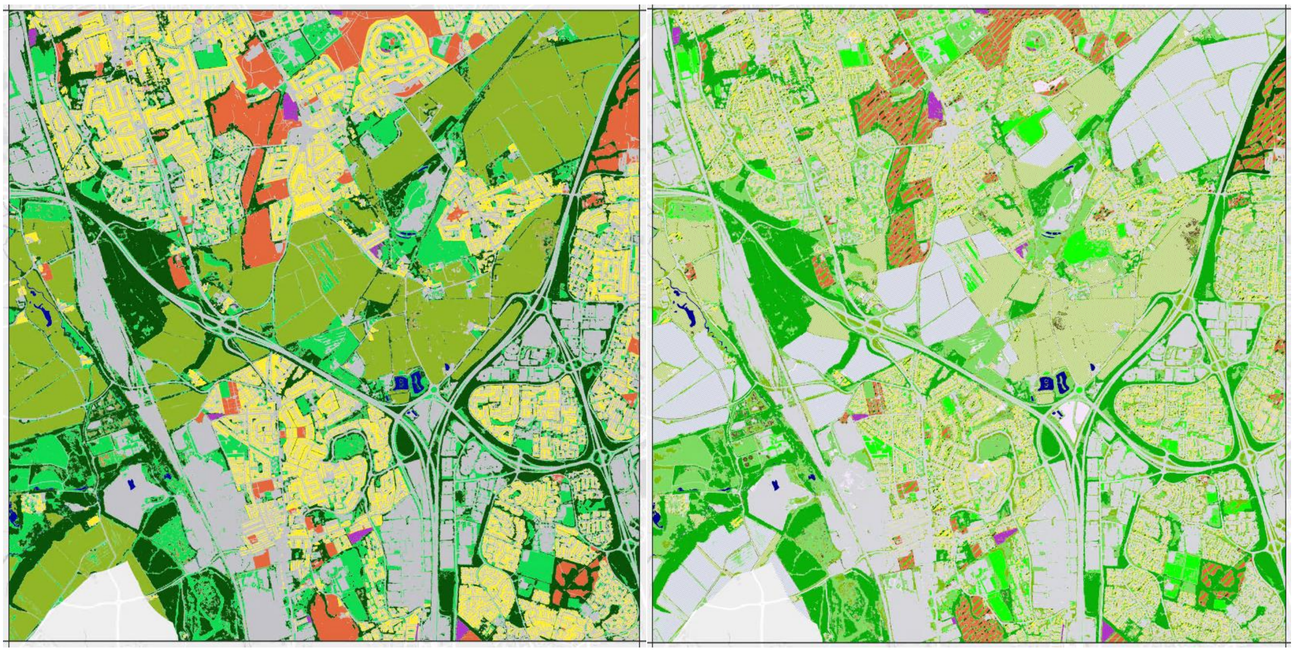


Figure 26. Example of a broad (left) and detailed (right) urban habitat map 5km OS grid tile in Tyneside.

Figure 27. Subset of the Tyneside aggregated naturalness map, where 1 is the highest level of Naturalness and 6 is the lowest.

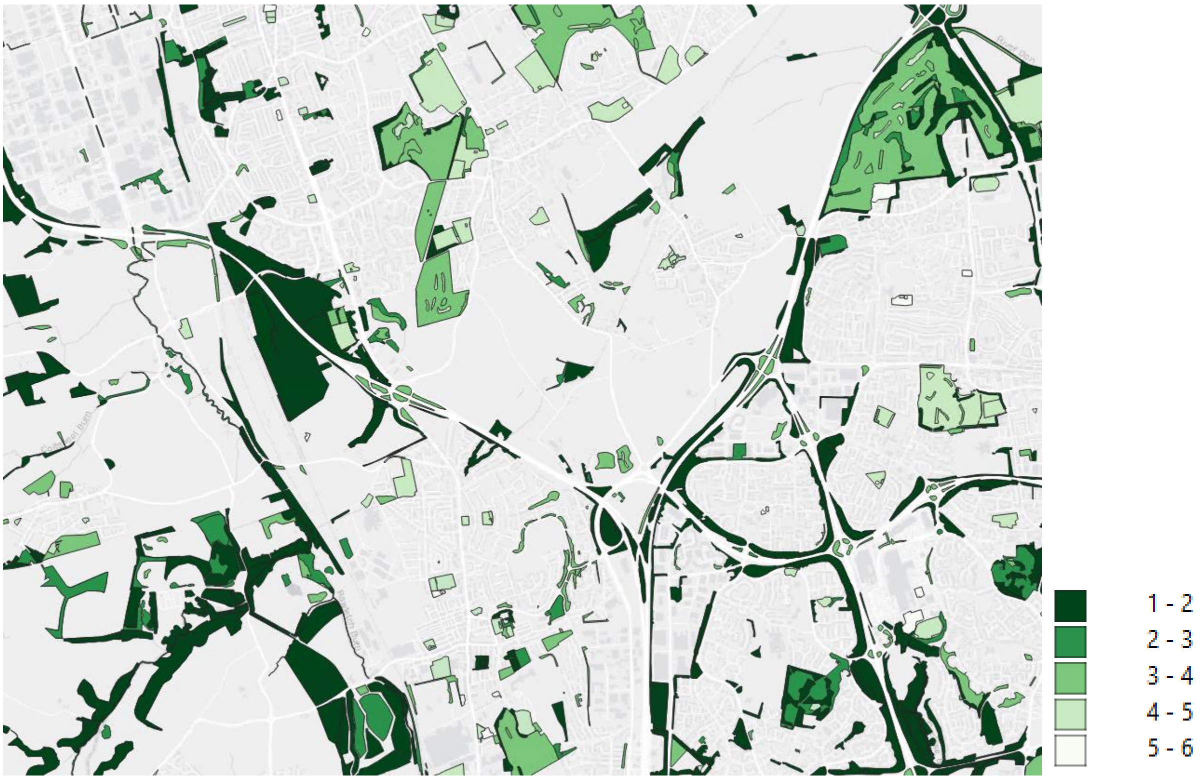


Figure 28. Subset of the Tyneside Green Infrastructure parcels, where 1-2 is the highest level of combined Naturalness factor and 5-6 is the lowest.

10. eCognition Issues and Potential Fixes

10.1. Error Loading Data in eCognition Server

If submitting an eCognition project to Server results in the error shown in **Figure 29**, the virtual mosaic files may need to be modified.

Job 83 Node 606: Back to overview Refresh

analyse - failed (00:00:00.03): Cambridge.tiles.tile23

Taken: 21.11.2022 18:57:22
Started: 21.11.2022 18:57:22
Finished: 21.11.2022 18:57:22

Engine: S2EG22HV02 / 21852 / 10.2.2 Build 130 x64 Log
Folder: \\S2EG22HV02\C$\Program Files\Trimble\eCognition GRID 10.2\bin\eCognitionEarthServer64.10.2.2.130\

Remarks:
Error loading data: PCI::OpenFile : Can't open '\\NAS2EGFB01-Data1\data\proj\GEO-PROJ-22-GIUP\WP2_Tiling\Data\APGB\Cambridge_APGB_CIR.vrt'. No valid PCI Driver found. Can't open '\\NAS2EGFB01-Data1\data\proj\GEO-PROJ-22-GIUP\WP2_Tiling\Data\APGB\Cambridge_APGB_CIR.vrt' with DIA GDAL driver v1.9.0 [GDAL 3.2.2, released 2021/03/05]: `R:/proj/GEO-PROJ-21-GREENINFRA/WP0_Data/APGB/CIR/113363-2_CIR_1_Shape/TL4159.jpg' does not exist in the file system, and is not recognized as a supported dataset name. (errorcode:2,2, reschedule) \\NAS2EGFB01-Data1\data\proj\GEO-PROJ-22-GIUP\WP2_Tiling\Data\APGB\Cambridge_APGB_CIR.vrt / S2EG22HV02 / 21852 / 10.2.2 Build 130 x64

Ruleset (client's path): \\NAS2EGFB01-Data1\data\proj\GEO-PROJ-22-GIUP\WP2_Tiling\Tiling_test\GIUP_ruleset_Cambridge_test.dcp
Process Name:

Figure 29. eCognition Server error showing an issue loading data from a virtual mosaic.

We recommend that users should replace all file paths containing a drive letter in the virtual mosaic files with the full path to the data storage, e.g., R:\ renamed to \NAS2EGFB01-Data1\data (**Figure 30**). This can be done in a text editor like Notepad ++.

```
<SimpleSource>
  <SourceFilename relativeToVRT="0">R:\temp\DataDir\APGB\CIR\GreaterManchester\SD5101.jpg</SourceFilename>
  <SourceBand>1</SourceBand>
  <SourceProperties RasterXSize="2000" RasterYSize="2000" DataType="Byte" BlockXSize="2000" BlockYSize="1" />
  <SrcRect xOff="0" yOff="0" xSize="2000" ySize="2000" />
  <DstRect xOff="0" yOff="36000" xSize="2000" ySize="2000" />
</SimpleSource>
<SimpleSource>
  <SourceFilename relativeToVRT="0">\\NAS2EGFB01-Data1\data\temp\DataDir\APGB\CIR\GreaterManchester\SD5101.jpg</SourceFilename>
  <SourceBand>1</SourceBand>
  <SourceProperties RasterXSize="2000" RasterYSize="2000" DataType="Byte" BlockXSize="2000" BlockYSize="1" />
  <SrcRect xOff="0" yOff="0" xSize="2000" ySize="2000" />
  <DstRect xOff="0" yOff="36000" xSize="2000" ySize="2000" />
</SimpleSource>
```

Figure 30. Top image: original file path pointing to the drive letter R; Bottom image: updated file path pointing to the data storage full path.

10.2. Error Processing LiDAR Tiles

If submitting an eCognition project to Server results in the error shown in **Figure 31**, it is likely that the merged LAZ point cloud for the given APGB date block is too large to be handled. This issue does not relate to constrains within the eCognition software, but rather to the computing power of the user, making the size threshold difficult to predict other than by trial and error. Based on the hardware specifications used by 2Excel[9], any APGB date block processing involving LAZ files equal to or greater than 5.37 GB failed. This roughly represents a surface area of 687 km^2.

[9] CPU: AMD EPYC 7713 @ 2GHz; GPU: 2 x NVIDIA RTX A5000; RAM: 1024 GB (DDR4); Storage: 2 TB (SSD); OS: Windows Server 22

```
[2023-01-11 12:28:48.593] S2EG22HV02 / 11252 : eCogSDEScanStreamCallback: Skipping wrong LAS index 428467794
[2023-01-11 12:28:48.593] S2EG22HV02 / 11252 : eCogSDEScanStreamCallback: Skipping wrong LAS index 428527612
[2023-01-11 12:28:48.593] S2EG22HV02 / 11252 : eCogSDEScanStreamCallback: Skipping wrong LAS index 428529343
[2023-01-11 12:28:48.593] S2EG22HV02 / 11252 : eCogSDEScanStreamCallback: Skipping wrong LAS index 428531775
[2023-01-11 12:28:48.593] S2EG22HV02 / 11252 : eCogSDEScanStreamCallback: Skipping wrong LAS index 428532234
[2023-01-11 12:28:48.593] S2EG22HV02 / 11252 : eCogSDEScanStreamCallback: Skipping wrong LAS index 428532258
[2023-01-11 12:28:48.593] S2EG22HV02 / 11252 : eCogSDEScanStreamCallback: Skipping wrong LAS index 428532263
[2023-01-11 12:28:48.593] S2EG22HV02 / 11252 : eCogSDEScanStreamCallback: Skipping wrong LAS index 428532276
[2023-01-11 12:28:48.593] S2EG22HV02 / 11252 : eCogSDEScanStreamCallback: Skipping wrong LAS index 428532280
[2023-01-11 12:28:48.593] S2EG22HV02 / 11252 : eCogSDEScanStreamCallback: Skipping wrong LAS index 428532288
[2023-01-11 12:28:48.593] S2EG22HV02 / 11252 : eCogSDEScanStreamCallback: Skipping wrong LAS index 428532297
[2023-01-11 12:28:48.593] S2EG22HV02 / 11252 : eCogSDEScanStreamCallback: Skipping wrong LAS index 428532305
[2023-01-11 12:28:48.702] S2EG22HV02 / 11252 : eCogSDEScanStreamCallback: Skipping wrong LAS index 428532314
[2023-01-11 12:28:48.702] S2EG22HV02 / 11252 : eCogSDEScanStreamCallback: Skipping wrong LAS index 428532320
[2023-01-11 12:28:48.702] S2EG22HV02 / 11252 : eCogSDEScanStreamCallback: Skipping wrong LAS index 428532330
[2023-01-11 12:28:48.702] S2EG22HV02 / 11252 : eCogSDEScanStreamCallback: Skipping wrong LAS index 428532338
[2023-01-11 12:28:48.702] S2EG22HV02 / 11252 : eCogSDEScanStreamCallback: Skipping wrong LAS index 428532339
[2023-01-11 12:28:48.702] S2EG22HV02 / 11252 : eCogSDEScanStreamCallback: Skipping wrong LAS index 428532340
[2023-01-11 12:28:48.702] S2EG22HV02 / 11252 : eCogSDEScanStreamCallback: Skipping wrong LAS index 428532342
[2023-01-11 12:28:48.702] S2EG22HV02 / 11252 : eCogSDEScanStreamCallback: Skipping wrong LAS index 428532343
[2023-01-11 12:28:49.732] S2EG22HV02 / 11252 : eCogSDEScanStreamCallback: Skipping wrong LAS index 428532718
[2023-01-11 12:28:49.732] S2EG22HV02 / 11252 : eCogSDEScanStreamCallback: Skipping wrong LAS index 428533129
[2023-01-11 12:28:49.732] S2EG22HV02 / 11252 : eCogSDEScanStreamCallback: Skipping wrong LAS index 428533529
```

Figure 31. eCognition Server error showing wrong LAS index.

If users encounter this issue, they should consider splitting up the problematic date block and making a note of the LAZ file size. The recommended method is as follows:

- Manually split up the 500m buffered AOI shapefile for the date block in a GIS software. You should do this by following the borders of the 5km OS grid tiles. We recommend adding a letter to the date in the naming convention to distinguish between the two new parts – this will not break the process. For example, splitting up the file GreaterManchester_20190526_AOI_500m_buffer.shp down the middle will result in:

  GreaterManchester_20190526**W**_AOI_500m_buffer.shp
  GreaterManchester_20190526**E**_AOI_500m_buffer.shp

- Use the new 500m buffered AOI parts to re-split the APGB, EA LiDAR, OSMM and GI datasets. This can be done manually in a GIS.
- Analyse both parts in eCognition like any other date block, following the usual process.

10.3. Other eCognition Related Issues

For any issues relating to the eCognition software, users can browse the eCognition Community website and utilise the forum: Topics – eCognition | Knowledge Base. Otherwise, Trimble support can be contacted at noreply@trimble.zendesk.com.

Appendix A

This appendix contains instructions on using the *prepare_apgb_by_date.py* Python script.

Function

This script is used to prepare the APGB datasets. These are downloaded as 1km tiles. The script finds all tiles belonging to the same collection date and merges them into a virtual mosaic (VRT). It results in a VRT file per APGB date block. It also exports a list of 1km OS grid codes associated with each APGB date block as a text file.

Assumptions

The script was written in Python 3.7. The following libraries are required for the script to function:

- os
- gdal (from osgeo)

The script assumes that the APGB 1km tiles for a given AOI have been downloaded and placed in folders as explained in **Section 3.4**.

Parameters

The script takes in four parameters. These should be modified by the user prior to processing and are described in **Table A1**.

Parameter	Data Type	Description
input_dir	String	Path to the data download directory, in the form of an r-string. This should exclude the last three folders (APGB, data type and AOI name)
data_type	String	Data type as a 3-letter code. This can be one of RBG, CIR, DTM or DSM
aoi	String	Name of the AOI. This is usually a Local Authority or Metropolitan County. The name should have no spaces, and upper-case letters should be used at the start of every word (e.g., GreaterManchester)
output_dir	String	Path to the root directory (**Section 4.1**) used to store prepared datasets, in the form of an r-string.

Table A1. Description of the four parameters used in the *prepare_apgb_by_date.py* script.

The parameters are used to construct a path to the input tiles, in the form {input_dir}\APGB\{date_type}\{aoi}, and to construct a path to the output virtual mosaics (VRT), as described in **Section 4.4** (**Table 3**), as well as to the output 1km OS grid list text files.

Appendix B

This appendix contains instructions on using the *create_apgb_date_blocks.py* Python script.

Function

This script is used to create the outlines of the APGB date blocks using the 500m buffered AOI. These are exported as ESRI Shapefiles.

Assumptions

The script was written in Python 3.7. The following libraries are required for the script to function:

- os
- glob
- geopandas

The script assumes that the text files containing the 1km OS grid tiles by APGB dates have been generated, and that the 500m buffered AOI (**Section 4.5**) and the OS grid tiles have been placed in the correct location (**Section 4.2**).

Parameters

The script takes in 2 parameters. These should be modified by the user prior to processing and are described in **Table B1**.

Parameter	Data Type	Description
aoi	String	Name of the AOI. This is usually a Local Authority or Metropolitan County. The name should have no spaces, and upper-case letters should be used at the start of every word (e.g., GreaterManchester)
data_dir	String	Name of the AOI. This is usually a Local Authority or Metropolitan County. The name should have no spaces, and upper-case letters should be used at the start of every word (e.g., GreaterManchester)

Table B1. Description of the 2 parameters used in the *create_apgb_date_blocks.py* script.

The parameters are used to construct a path to the split 500m buffered AOI tiles, in the form {data_dir}\AOI\{aoi}\{date_block}_AOI_500m_buffer.shp, as described in **Section 4.5** (**Table 4**).

Appendix C

This appendix contains instructions on using the *prepare_ea_lidar.py* Python script.

Function

This script is used to prepare the EA LiDAR dataset from the EA National LiDAR Programme. These are downloaded as 5km tiles. The script finds all tiles belonging to the same APGB date block and merges them into a GeoTIFF (TIF) or point cloud (LAZ), depending on the input data type. It results in a TIF or LAZ file per APGB date block.

Assumptions

The script was written in Python 3.7. The following libraries are required for the script to function:

- os
- gdal (from osgeo)
- zipfile
- subprocess

LASTools should be available for use. The path to the LASTools directory can be specified by the user in the script parameters.

The script assumes that the EA LiDAR DTM 5km tiles for a given AOI have been downloaded and placed in folders as explained in **Section 3.8**. It also assumes that the APGB imagery has been prepared as described in **Section 4.4** and that the 1km OS grid code lists have been generated as text files (**Appendix A**). These are a requirement.

Parameters

The script takes in five parameters. These should be modified by the user prior to processing and are described in **Table C1**.

Parameter	Data Type	Description
input_dir	String	Path to the data download directory, in the form of an r-string. This should exclude the last three folders (EA_LiDAR, date type and AOI name)
data_type	String	Data type as a 3-letter code. This can be one of DTM, DSM or LAZ.
aoi	String	Name of the AOI. This is usually a Local Authority or Metropolitan County. The name should have no spaces, and upper-case letters should be used at the start of every word (e.g., GreaterManchester)
output_dir	String	Path to the root directory (**Section 4.1**) used to store prepared datasets, in the form of an r-string.
lastools_dir	String	Path to the LASTools directory, in the form of an r-string.

Table C1. Description of the five parameters used in the *prepare_ea_lidar.py* script.

The parameters are used to construct a path to the input tiles, in the form {input_dir}\EA_LiDAR\{data_type}\{aoi}, and to construct a path to the output files (TIF or LAZ), as described in **Section 4.8** (**Table 7**).

Appendix D

This appendix contains instructions on using the *generate_confusion_matrix.py* Python script.

Function

This script is used to generate a confusion matrix for a given eCognition output tile, and to calculate the overall accuracy value of the detailed urban habitat classification. The script provides the option to save the confusion matrix as an image (**Figure 25**).

Assumptions

The script was written in Python 3.7. The following libraries are required for the script to function:

- os
- numpy
- geopandas
- sklearn.metrics
- matplotlib.pyplot

The script assumes that the given tile has been assessed using a desk-based survey technique as explained in **Section 7**.

Parameters

The script takes in 1 required parameter and 2 optional ones. It should be modified by the user prior to processing and is described in **Table D1**.

Parameter	Data Type	Description
input_file	String	Path to the tile to be assessed, in the form of an r-string.
predicted_attribute	String	Attribute field name containing predicted classes from the workflow. Default is 'Detailed_H'.
actual_attribute	String	Attribute field name containing the actual classes from the desk-based assessment. Default is 'Actual'.

Table D1. Description of the 1 required parameter and the 2 optional ones used in the *generate_confusion_matrix.py* script.

Appendix E

This appendix contains instructions on using the *rename_gdb_tiles.py* Python script.

Function

This script is used to rename urban habitat classification 5km geodatabase (GDB) tiles that are created by the eCognition software to a more usable format:

{aoi}_{date}_{os_grid}_Urban_Habitat_Map.gdb

The files are not moved and are renamed within the existing folder.

Assumptions

The script was written in Python 3.7. The following libraries are required for the script to function:

- os

The script assumes that the eCognition tile outputs follow the naming convention described in **Section 6.5.4**:

{aoi}_{date}.Copies.subsets.{os_grid}_Classification.gdb

Parameters

The script takes in 1 parameter. It should be modified by the user prior to processing and is described in **Table E1**.

Parameter	Data Type	Description
input_dir	String	Path to the directory containing the output GDB tiles, in the form of an r-string.

Table E1. Description of the 1 parameter used in the *rename_gdb_tiles.py* script.

The input parameter also dictates the output directory, as the tiles are renamed but not moved.

Appendix F

This appendix contains instructions on using the *calculate_combined_naturalness.py* Python script.

Function

This script is used to merge all GI parcels created by the eCognition software across all date blocks, and generates a new GeoPackage in the user's chosen directory:

{aoi} - Green Infrastructure Parcels.gpkg

There is an input CSV for every 5km OS grid tile in each of the date blocks, and for every single TypologyType (inherited from the GI database).

Assumptions

The script was written in Python 3.7. The following libraries are required for the script to function:

- os
- glob
- pandas
- geopandas

The script assumes that the eCognition GI parcels CSV files follow the naming convention described in **Section 6.5.4**:

{aoi}_{date}.Copies.subsets.{os_grid}_{TypologyType}.csv

It also assumes that the GI database has been prepared as explained in **Section 4.6**.

Parameters

The script takes in 4 parameters. These should be modified by the user prior to processing and are described in **Table F1**.

Parameter	Data Type	Description
aoi	String	Name of the AOI. This is usually a Local Authority or Metropolitan County. The name should have no spaces, and upper-case letters should be used at the start of every word (e.g., GreaterManchester)
ecog_workspace	String	Path to the eCognition workspace directory, in the form of an r-string.
data_dir	String	Path to the root directory (**Section 4.1**) used to store prepared datasets, in the form of an r-string.
output_dir	String	Path to the output directory containing the final datasets, in the form of an r-string.

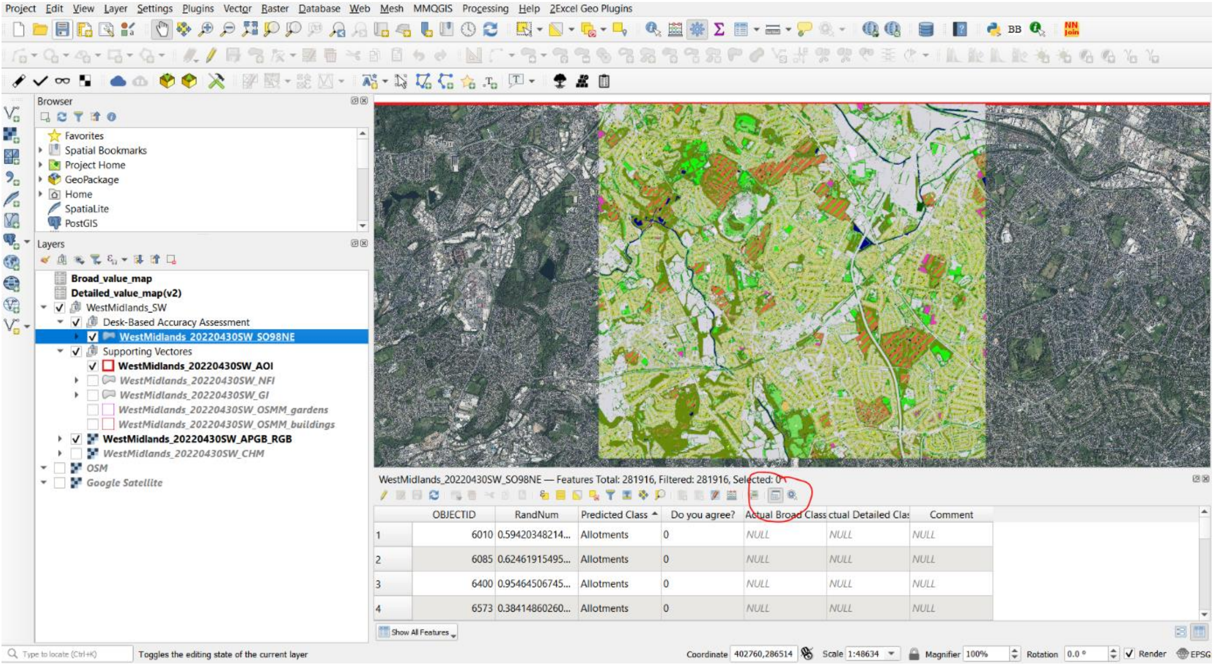
Table F1. Description of the 4 parameters used in the *calculate_combined_naturalness.py* script.

The parameters are used to construct a path to the input CSV files, in the form {eCognition workspace}\results\GI_Parcels\{aoi}_*.csv, and to construct a path to the output GeoPackage (GPKG), as described in **Section 8.3**.

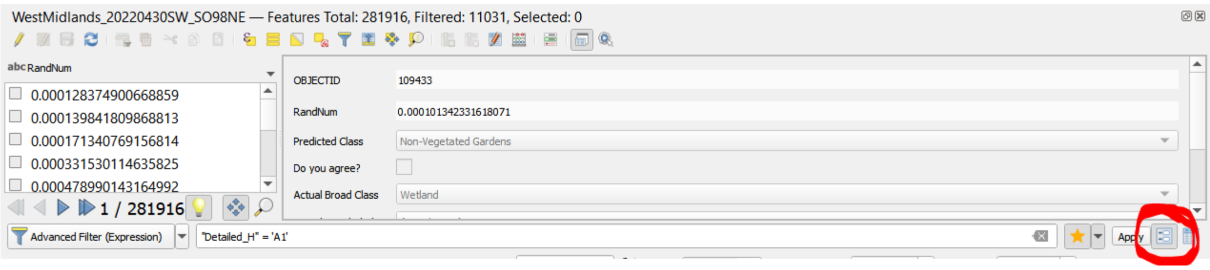
Appendix G

<u>Set Up</u>

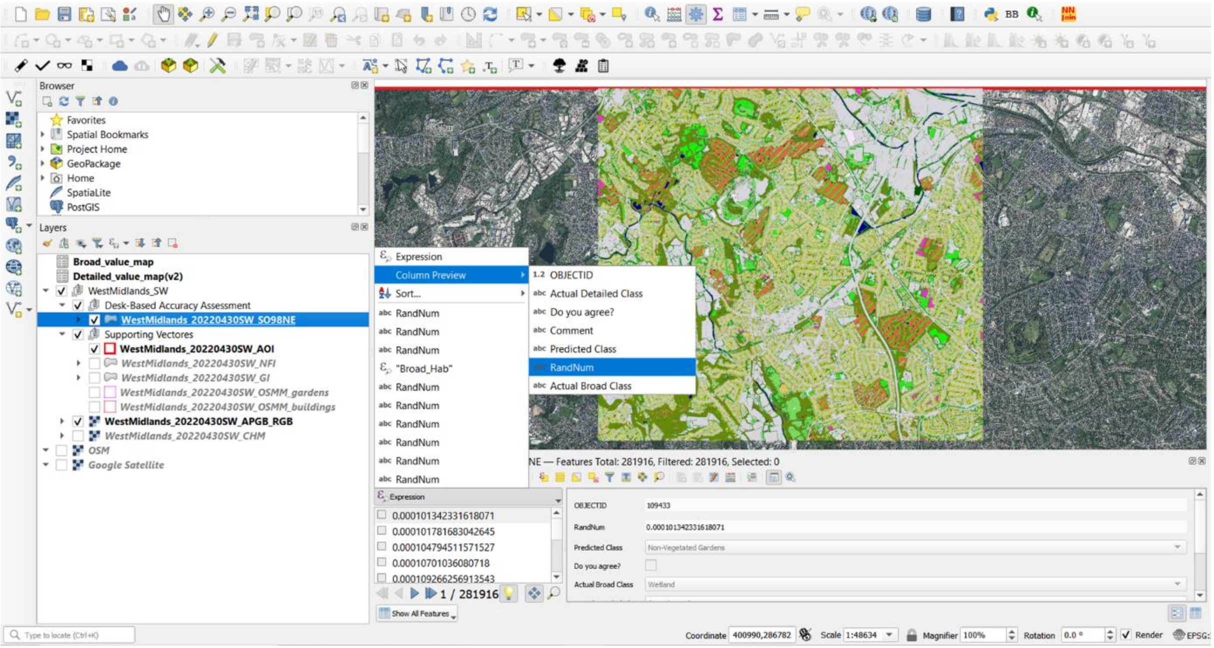
1. Open the attribute table of the Desk-Based Accuracy Assessment shapefile and make sure to **"Dock Attribute Table"** (red circle).



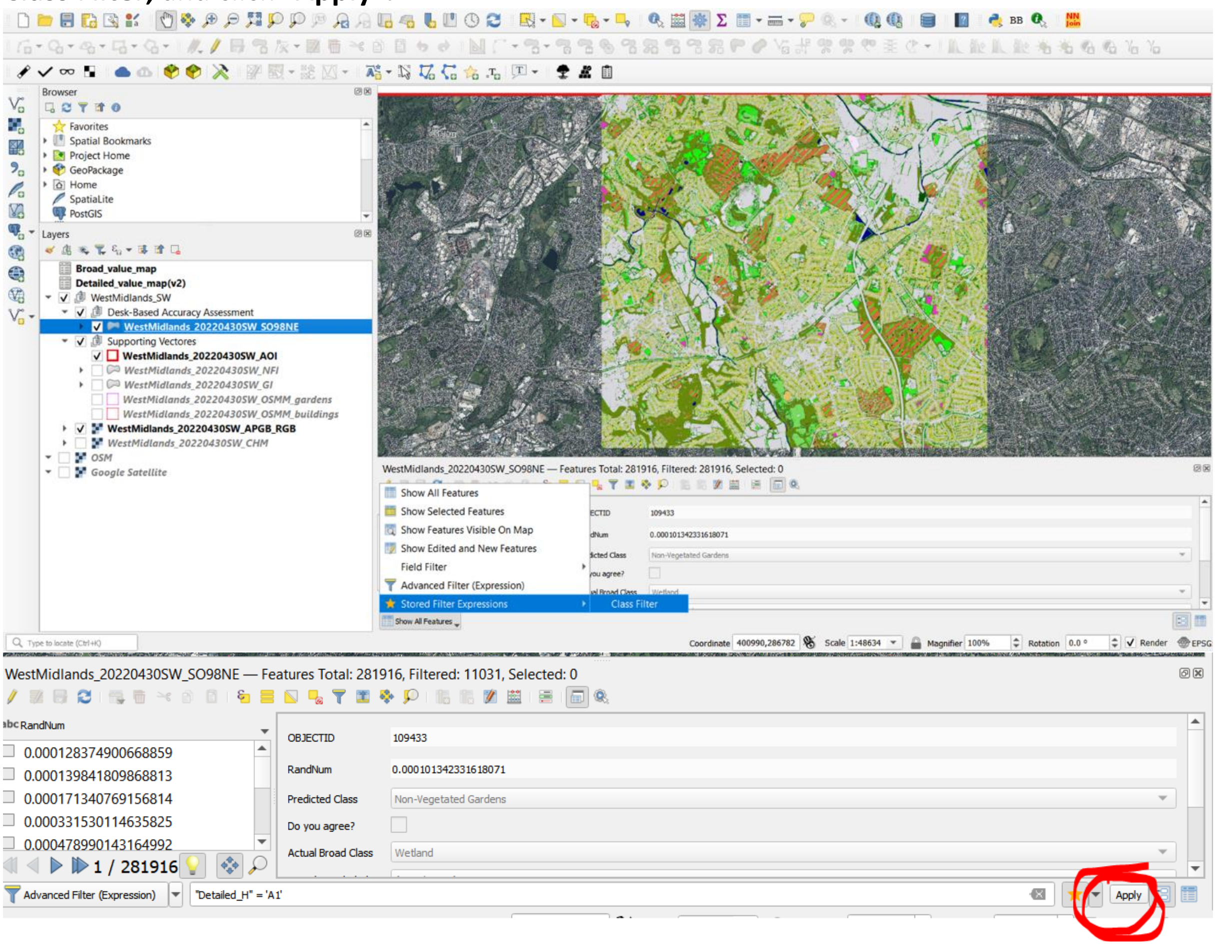
2. Click on **"Switch to form view"**.



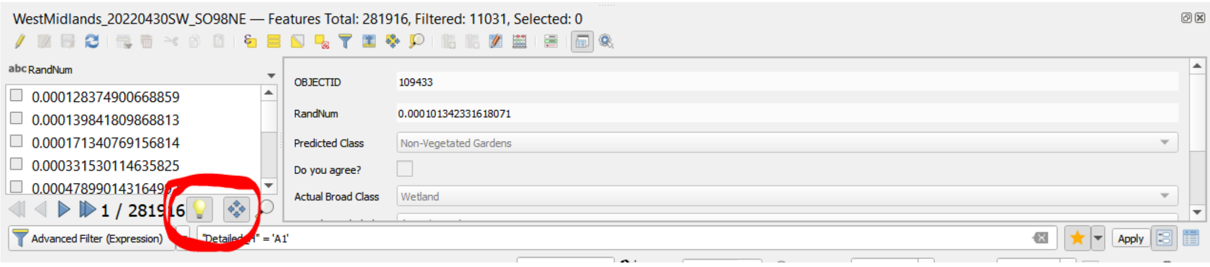
3. On the left of the attribute table, click on “**Expression**” → “**Column Preview**” and select the



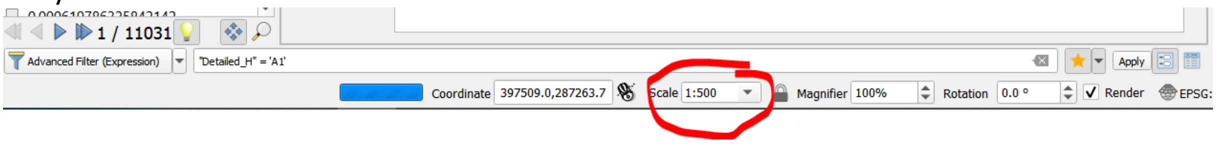
4. In the lower left corner, click on “**Show All Features**” → “**Stored Filter Expressions**”, select **Class Filter,** and click **“Apply”.**



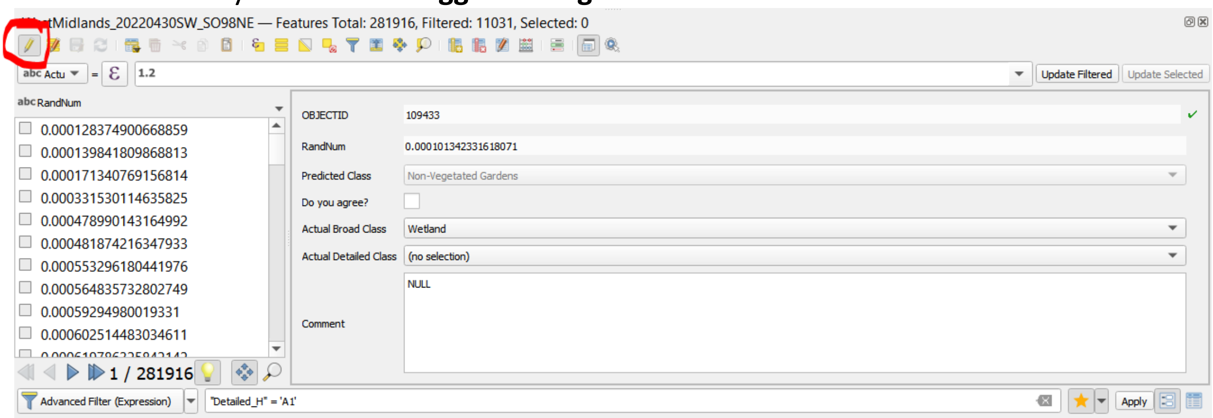
5. Make sure both the "**Highlight current feature on map**" and the "**Automatically pan to the current feature**" options are selected.



6. Set your scale to 1:500.

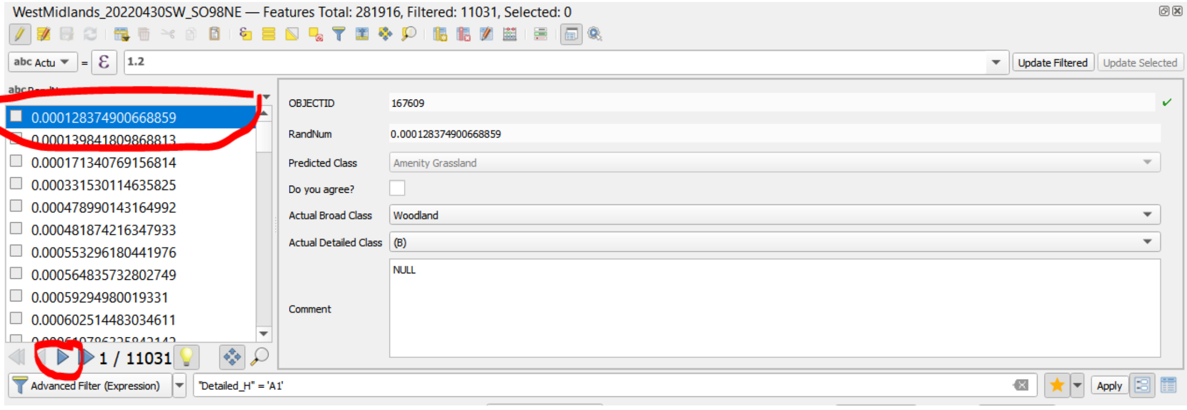


7. You are now ready to turn on "**Toggle editing mode**" and start the assessment.

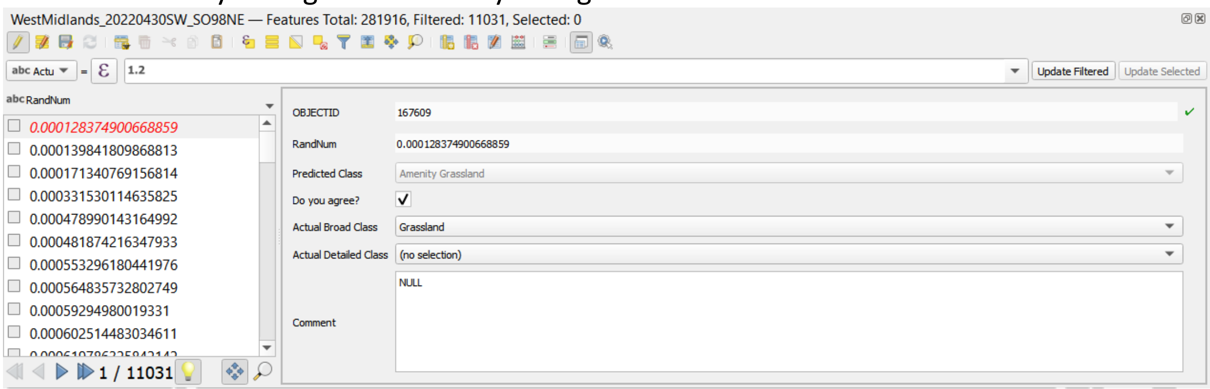


Assessment

1. Click on the first item on the list, and then use the blue arrow to move to the next one. The chosen object should flash red. You can click the flashlight "**Highlight current feature on map**" button on and off to see it flash again.

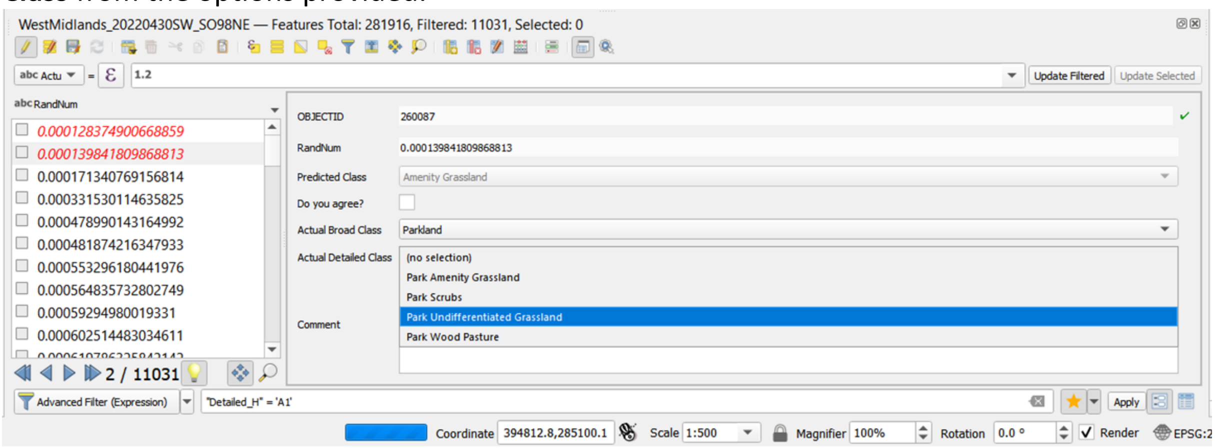


2. **If you agree** with the **Predicted Class**, check the **"Do you agree?"** box. You don't have to do anything else, and you can move on to the next object. The **Actual Broad Class** and **Actual Detailed Class** fields will be filled in automatically. If you're using your mouse to scroll, be careful because you might accidentally change the values in those fields.



If the **Actual Detailed Class** says "(no selection)" after you ticked the box, it was in fact filled in.

3. **If you disagree** with the **Predicted Class**, select the **Actual Broad Class** and **Actual Detailed Class** from the options provided.



4. Please leave comments in the **Comment** field if you are unsure about you assessment, or spotted something strange/interesting.

5. Assess a total of **10 samples** of a class, then change the class key code in the Advanced Filter (expression) to the next detailed class and hit "**Apply**".

6. Make sure to **save** your changes frequently!