

Segmenting Urban Green Spaces in Satellite Images using Deep Reinforcement Learning

Isaac Wolf

2009257

Submitted to Swansea University in fulfilment
of the requirements for the Degree of Master of Science



Swansea University
Prifysgol Abertawe

Department of Computer Science
Swansea University

May 6, 2025

Declaration

This work has not been previously accepted in substance for any degree and is not being concurrently submitted in candidature for any degree.

Signed *Isaac Wolff* (candidate)

Date 06/05/2025

Statement 1

This thesis is the result of my own investigations, except where otherwise stated. Other sources are acknowledged by footnotes giving explicit references. A bibliography is appended.

Signed *Isaac Wolff* (candidate)

Date 06/05/2025

Statement 2

I hereby give my consent for my thesis, if accepted, to be available for photocopying and for inter-library loan, and for the title and summary to be made available to outside organisations.

Signed *Isaac Wolff* (candidate)

Date 06/05/2025

Thank you to Mum and Dad for all your support.

Abstract

Mapping Urban Green Spaces (UGS) from satellite imagery is an essential component for urban planning and environmental monitoring, but generating precise pixel-level annotations for training semantic segmentation models can be resource intensive and costly. Standard datasets frequently suffer from significant class imbalance, with target classes being dominated by background pixels and therefore end up being under-represented. This project addresses these challenges by developing and evaluating a framework designed to reduce annotation effort while maintaining sufficient segmentation performance.

To achieve this, we leverage two independent binary segmentation models (BSMs), based on the DeepLabV3+ architecture with a ResNet18 backbone, which are trained to identify green space and urban area pixels respectively. These models are trained and evaluated on the Five-Billion-Pixels (FBP) dataset, which offers a diverse, high-fidelity and multi-spectral benchmark. To minimise labelling requirements for these models, we employ an Active Learning (AL) strategy where two corresponding Deep Q-Networks (DQNs) agents, for each BSM, learn policies to select the most informative image tiles for labelling. The agents are guided by a state representation that captures the model's current performance, uncertainty/entropy score, and training progress. During training, class imbalance in the FBP dataset is addressed using a weighted cross-entropy loss function that compensates for the large proportion of background pixels. We then provide tile-level UGS labels through manual labelling at the tile-level, creating expert definitions of UGSs. A dedicated Tile Classifier Model (TCM), using a CNN architecture, is then trained to predict these expert labels based on the outputs generated by the BSMs. We hypothesise that this segmentation process could reduce the manual annotation effort required for UGS mapping, while incurring minimal impact to the segmentation performance, and therefore serve as an alternative approach to the standard satellite segmentation methods that exist.

Contents

List of Tables	v
List of Figures	vi
1 Introduction	1
2 Related Work	3
3 Dataset	7
3.1 Multi-Spectral Data	7
3.2 Annotations	8
4 Methodology	11
4.1 Data Preprocessing	11
4.2 Segmentation Model	13
4.3 Active Learning Framework	14
4.4 UGS Definition from Expert Labelling	18
4.5 UGS Tile Classifier Model	18
4.6 Evaluation	19
4.7 Active Learning Algorithm	21
5 Experiments	23
5.1 Hardware Specifications	23
5.2 Hyper Parameters	23
5.3 ResNet Depth	26
6 Results	27
6.1 Quantitative Metrics	28
6.2 Qualitative Metrics	30
7 Discussion	33

8 Conclusion	35
Bibliography	37
Appendices	41
A Dataset Statistics	43
B Additional Metrics	45

List of Tables

B.1	Exact metric values showing comparison of different ResNet backbones on segmentation performance	45
B.2	Metrics for the green space, urban area and UGS evaluations when randomly selecting tiles	46
B.3	Metrics for the green space, urban area and UGS evaluations when selecting tiles with the AL framework	46

List of Figures

3.1	Five-Billion-Pixels dataset tile examples and class legend	8
3.2	Distribution of pixels across our three class definition in the Five-Billion-Pixels dataset	9
4.1	Example tile that has been preprocessed ready for training with	12
4.2	DeepLabV3+ Architecture	13
5.1	Segmentation model learning rate tuning tests	25
5.2	Tile size performance testing	25
5.3	Comparison of different ResNet backbones on segmentation performance	26
6.1	Comparison of evaluation metrics for the random tile selection baseline benchmark .	28
6.2	Comparison of evaluation metrics for the AL tile selection framework	28
6.3	Comparison of evaluation metrics between the baseline and AL framework as a radar graph	29
6.4	Alpha values for the green space and urban area models	29
6.5	Q-values for the green space and urban area models	30
6.6	Four example tile predictions that show both mostly correct and over-predicted UGS classifications	31
6.7	Full image UGS prediction visualisation made from a composition of tile predictions	32
A.1	Distribution of pixels across the 24 classes in the Five-Billion-Pixels dataset	43

Chapter 1

Introduction

Urban Green Spaces (UGSs) are commonly described as open spaces in cities such as forests and parks, which help to enhance sustainability and urban habitability in cities. Studies show that UGSs can help reduce air pollution in major cities, an urgent sustainability issue recognised by the UN [1], along with displaying mental and physical health benefits for urban populations [2]. Research additionally demonstrates that UGSs can reduce the effects of urban heat islands in cities, creating higher quality spaces for residents [3]. As rapid urbanization is on the rise, it has become more vital than ever that the management of UGSs is handled using robust and intelligent tools that can accurately map satellite images to detect the presence or absence of these spaces.

High spatial resolution satellite images allow us to monitor the data of massive landscapes, making for a powerful resource when monitoring changes in our environment [4, 5]. However, the task of segmenting UGSs from satellite images comes with a number of challenges. These challenges include tackling the large variation in vegetation types, which are further complicated by seasonal changes, and discerning between the mix of both man-made and natural features in urban environments [3, 5].

Existing methods can be limited by high computational costs, class imbalances, and the need for large quality labelled datasets to train on [6], which is where this project's implementation of an Active Learning (AL) approach, that uses a Deep Q-Network (DQN) architecture, aims to improve upon. We hypothesise that by combining AL with a DQN, the human input and computational costs could be reduced through reward-based learning, as this type of learning allows models to better adapt to novel environments [6]. The model works by exploiting areas of the image that provide the highest rewards, and ignoring less useful regions, which is explained in more depth in Chapter 4.

Contributions to satellite image segmentation could help innovations in environmental monitoring and natural disaster prevention by allowing accurate, real-time analysis of large scale landscape data. An example of this can be seen in Tupayachi's paper [7] which analyses satellite images using DRL to combat wildfires in real time. Additionally, an important observation made by Jodeiri Rad and Armenakis' paper [6] is that by improving our understanding of satellite im-

1. Introduction

age segmentation, we can further our knowledge in other fields such as medical image analysis, autonomous driving and augmented reality. This serves as a solid motivation for attempting to experiment with techniques such as AL and DQNs, which have seen emerging applications in those fields.

Lastly, please note that throughout this paper the terms 'AL agent' and 'DQN agent' are used interchangeably, and refer to the same agent.

Chapter 2

Related Work

In this section, we cover the diverse number of use cases for RL spanning across fields such as urban planning tasks, video segmentation, and medical imaging. Here we will examine the types of methods used by relevant papers in this field, and analyse some of the limitations that projects can face when leveraging RL techniques and creating capable satellite segmentation models.

Casanova's paper [8] introduces an active learning strategy aiming to minimise annotation costs when performing semantic segmentation. The paper's suggested model focuses on urban street views rather than satellite images and employs a modified DQN that selects from informative regions rather than the entire image, helping to optimise the annotation process. Their paper makes a key contribution by addressing the problem of class imbalances in the dataset by prioritising under-represented classes, namely bicycles and pedestrians over the more common classes of roads, buildings and sky. Their final framework managed to achieve a 30% reduction in labelling requirements compared to state-of-the-art baselines [8] while also achieving comparable performance when segmenting the Cityscapes dataset¹. However, the paper's suggested method of keeping the image region size fixed for every sample that it segments may limit the model's adaptability when segmenting novel datasets. Therefore, exploring a more flexible region size selection technique that dynamically adjusts the image tiling size based on the uncertainty of pixel classifications within the image could help to improve the model's performance, and is a method our project may look to employ.

Similarly to Casanova's paper, Jodeiri Rad and Armenakis' paper [6] investigates DRL in semantic segmentation, and focuses on urban imagery where data labelling is highly resource intensive. Using an active learning method with a Deep Q-Network and Frequency Weighted Average Intersection Over Union IoU (IoU) metric, which will be discussed in more depth in the methodology Section 4, the model demonstrates the ability to achieve a nearly fully supervised model accuracy of 98% by labelling only 8% of the dataset [6]. This approach shows an impressive balance of computational efficiency and segmentation accuracy. One criticism that could be

¹<https://www.cityscapes-dataset.com/>

2. Related Work

made is that the paper does not present the results of any alternative metrics to the Frequency Weighted IoU calculation, and therefore may be missing out on other strategies such as recall and dice coefficient scores, that might address the class imbalance and data scarcity issues in a more effective way.

One study that researches the segmentation of UGSs is Huerta's paper [5] which outlines how very high-resolution satellite imagery and deep learning can be used to segment and classify polygons of UGSs at a street view scale. However, instead of using DRL like our project is suggesting, the paper solely employs a convolutional neural network (CNN), specifically a U-Net architecture with ResNet encoders, to handle the UGS segmentation task. The paper addresses the class imbalance issues that are common in satellite segmentation tasks, where non-UGS areas dominate the dataset, through data augmentation techniques and additional spectral bands such as Near-Infrared (NIR), to produce a more adaptable model that can learn effectively in novel environments [5].

Building on CNN-based segmentation for UGS mapping [5], recent research has also experimented with alternative architectures for complex Land Cover Classification (LCC) using Multispectral Imagery (MSI). One such research project is Ramos' paper [9] that takes advantage of SK-ResNeXt as its backbone, it claims to further improve performance by adapting to spatial and spectral complexities seen in satellite imagery. Similar to our project, the paper evaluates its model on the Five-Billion-Pixels dataset [4]. They reported that their U-Net + SK-ResNeXt50 architecture achieved over 5% higher accuracy and 7% higher mean IoU compared to baseline models, particularly when using RGB-NIR data [9]. Visual tests confirmed better feature delineation, highlighting the model's intelligent use of NIR information [9]. The paper shows integrating SK-ResNeXt can offer a new high performance alternative to the standard DeepLabV3+ and ResNet backbone [10], that our project chose to utilise.

Reinforcement learning has also recently seen use in some medical fields, as evidenced by Hu's 2022 paper [11] which describes how RL can be applied to medical image analysis, displaying its ability to perform segmentation tasks across procedures such as magnetic resonance imaging (MRI), ultrasound, and computed tomography (CT) scans. The paper demonstrates how RL methods excel in sequential decision-making, allowing them to be effective for tasks requiring an iterative improvement process. The paper also displays RL's advantages in minimising labelling biases and handling sequential data by not relying too much on manually labelled datasets and instead using the RL agent's ability to learn by interacting with an environment and receiving feedback in the form of rewards. In spite of the exciting results shown, the paper identifies multiple issues to do with the scarcity of annotated datasets and high computational demands that the medical field often requires. The study suggests implementing transfer learning and domain adaptation techniques to overcome these problems, and to aid in improving RL's application to healthcare challenges [11].

Other studies that are relevant to this project include Srikishan's paper [12], discussing the

use of a Parsimonious Segmentation with Reinforcement Learning (PaSeR) framework which optimises their model’s performance through a IoU/GigaFlop metric. This metric examines how the performance of a task affects the computational expenditure. Their suggested PaSeR design demonstrates a sizeable increase in the model’s efficiency, particularly with the battery material phase segmentation task, where resources were limited. Another relevant study is Zeng’s paper [13], which uses RL to segment images of gold immunochromatographic strips (GICS) used in metal and ore analysis. Their novel approach was to introduce a deep belief network (DBN) into the existing DQN framework. This showed improvements in both the model’s efficiency and accuracy when segmenting GICS images. Additionally, they included a multi-factor learning curve, which was shown to be particularly effective for low contrast and noisy images.

Chapter 3

Dataset

This project uses the Five-Billion-Pixels (FBP) dataset [4] as its source of training and evaluation information. The FBP dataset is an extensive benchmark created for semantic segmentation tasks in satellite observation that uses high-resolution remote sensing data. This dataset is comprised of 150 images, each approximately 7000 x 7000 pixels in size, and captured by the Gaofen-2 (GF-2) satellite¹. These images span over 50,000 km² across more than 60 districts within China, ensuring a wide geographical distribution that reflects variations in climate, altitude, and geology. The dataset's scale, resolution, and detailed annotations, present a valuable chance to test modern deep learning approaches that seek to improve satellite segmentation quality.

3.1 Multi-Spectral Data

The GF-2 satellite is equipped with Panchromatic and Multispectral sensors, and is a part of the High-Definition Earth Observation System program by the China National Space Administration [4]. GF-2 can capture multispectral imagery at a spatial resolution of four meters, and features a resolution depth of 16 bits per channel, offering a wider dynamic range of intensity values when compared to standard 8-bit imagery [4]. This high-fidelity supports capturing variations in surface reflectance and distinguishing between urban and natural features relevant to UGS analysis [14].

Each image file is comprised of four spectral bands with the following characteristics [4]:

- **Blue (B):** 0.45-0.52 μm . Assists with atmospheric correction and water body penetration.
- **Green (G):** 0.52-0.59 μm . Displays any chlorophyll absorption and provides contrast with the NIR band for vegetation.
- **Red (R):** 0.63-0.69 μm . Corresponds to a chlorophyll absorption and provides contrast with the NIR band for vegetation analysis.

¹<https://www.eoportal.org/satellite-missions/gaofen-2>

- **Near-Infrared (NIR):** 0.77-0.89 μm . Is highly sensitive to vegetation cell structure and water content, exhibiting higher reflectance for healthy vegetation compared to most non-vegetated surfaces. The characteristics of this NIR band permit the calculation of vegetation indices such as EVI (Enhanced Vegetation Index), discussed in Section 4.1.

This combination of a four-meter spatial resolution and multiple spectral bands, helps to map UGSs from complex urban and peri-urban landscapes [4, 15].

3.2 Annotations

The FBP dataset provides dense, pixel-level ground truth annotations, which were obtained through careful manual labelling with multiple checking phases to ensure consistency and accuracy [4].

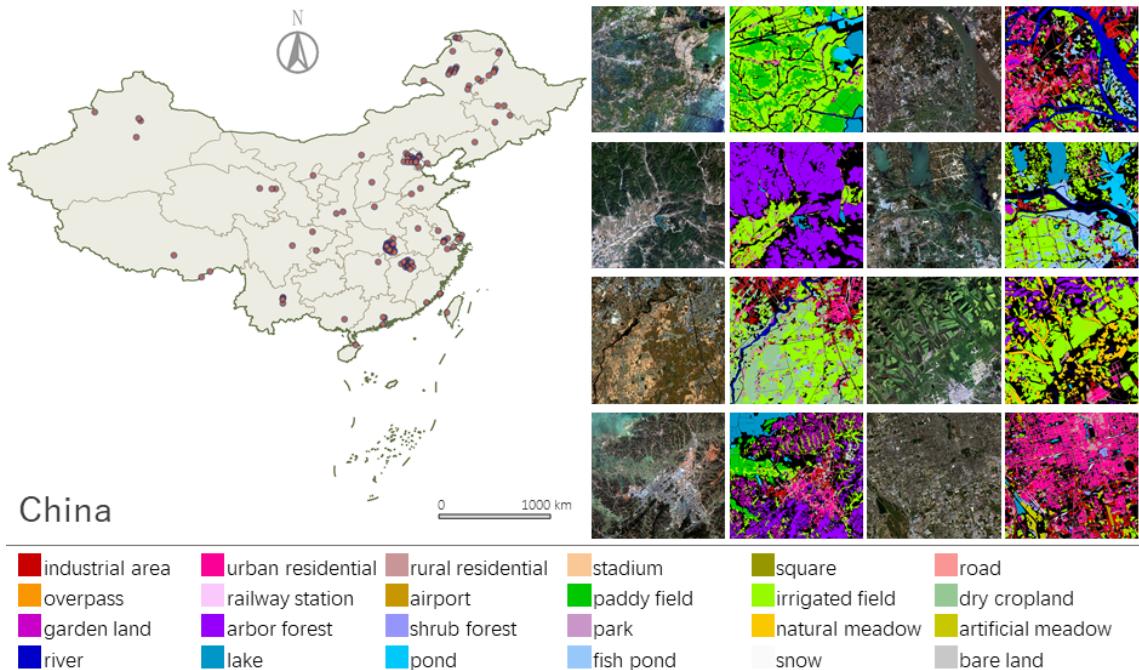


Figure 3.1: This image is taken from the FBP paper [4]. It shows a basic map of the areas that the dataset covers (the top left image). It also shows examples of both the BGRNir images and their corresponding 24 class masks (the top right image). The legend for the 24 class mask is displayed at the bottom of the image.

The masks contain 24 unique classes that cover artificial, agricultural, and natural categories, specifics of which are listed in the legend of Figure 3.1. Additionally, a value is reserved for unlabelled pixels representing miscellaneous or unclear areas [4]. This comprehensive category system, including sub classifications of urban areas, e.g., stadiums, squares, and parks, facilitates analysis beyond basic land cover.

For our segmentation task, we can simplify the image mask into three broader categories: background, green space, and urban area. We make this simplification during the data preprocessing stage, which is discussed in Section 4.1. The original 24-class annotations provide a useful

high-fidelity ground truth, and guarantees that the derived green space and urban area labels are based on an accurate representation of the underlying land cover.

Lastly, the FBP dataset exhibits a considerable class imbalance. For our three-class definition, the background class contribute to approximately 78% of all pixels, while the target/positive classes, green space and urban area, represent roughly 10% and 12%, respectively. This is detailed in the Figures below 3.2, and the exact 24 class pixel distribution breakdown can be found in the Appendix A. This imbalance poses as an obstacle for segmentation models, which can become biased towards the dominant background class [16]. We discuss how this is handled in Section 4.2.3.

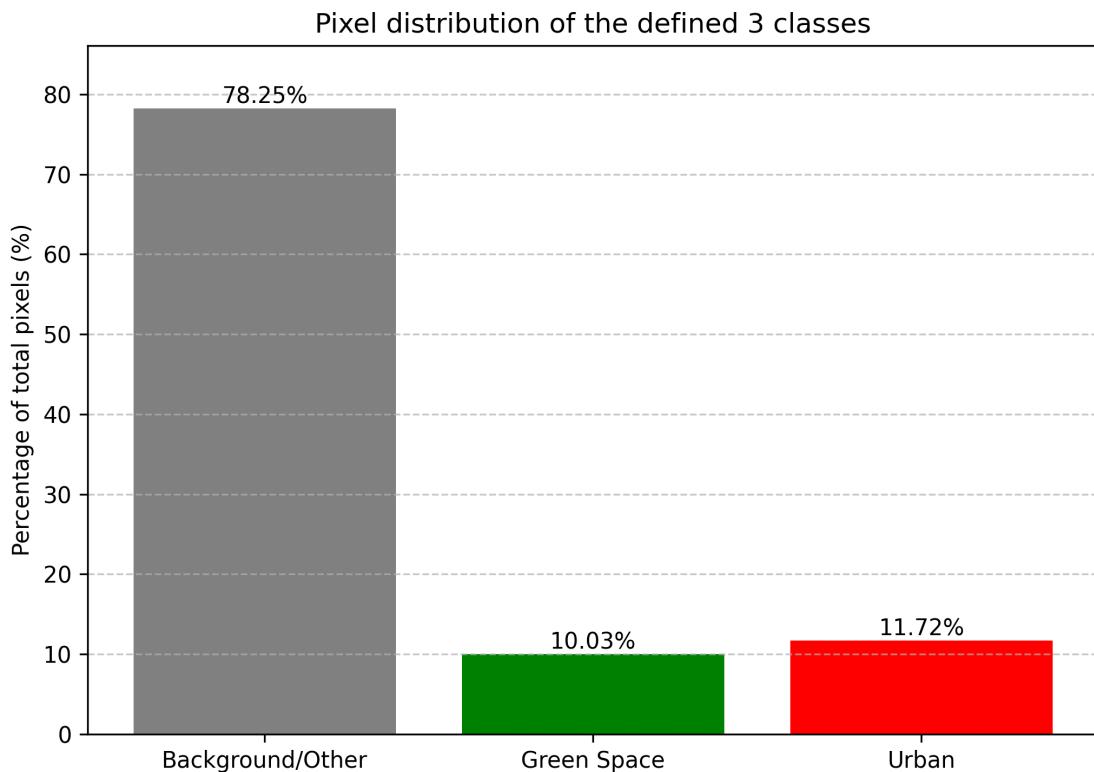


Figure 3.2: This shows the pixel distribution across the three classes we defined from the FBP dataset [4]. In our definition, we can see that only around 22% of the dataset (green space + urban area distribution) actually consists of the desired pixels that we want to segment dataset.

Chapter 4

Methodology

This chapter explains the methodologies we employ to create our segmentation models and AL agents. To first gain a high level understanding of the workflow, we can outline a brief description of the overall approach, shown as a multi-stage process:

1. First, we train independent two Binary Segmentation Models (BSMs) to semantically segment our green space and urban area definitions. This is done using the AL framework to improve annotation costs.
2. We manually label tiles to store expert knowledge about ground truth UGSs at the tile level.
3. Using the expert knowledge, we train a Tile Classifier Model (TCM) that learns to predict UGSs in the outputs from the BSMs.
4. The system is evaluated by combining the predictions from the BSMs, and then classifying any resulting UGSs using the TCM.

4.1 Data Preprocessing

Prior to training, we first process the raw 16-bit BGRNir images from the FBP dataset into tiles that are ready to be used for training. Initially, each large image is divided into non-overlapping tiles of size 512×512 pixels. For each tile, we calculate the Enhanced Vegetation Index (EVI) using the standard EVI formula seen in Equation 4.1, and add it as an additional channel, resulting in a final five channel set (B, G, R, NIR, EVI):

$$EVI = G \times \frac{NIR - Red}{NIR + C_1 \times Red - C_2 \times Blue + L} \quad (4.1)$$

The exact constants used during processing are $G = 2.5$, $C_1 = 6$, $C_2 = 7.5$, and $L = 1$, as per the MODIS Vegetation Index User's Guide [17].

Following the raw 16-bit image processing, we process the corresponding mask file. This involves converting the image's original 24-class FBP annotated mask, seen in Figure 3.1, into a

4. Methodology

three class semantic mask. These three classes represent background, green space, and urban area. The exact splitting of classes is as follows:

- **Background** - unlabelled, paddy field, irrigated field, dry cropland, river, pond, fish pond, snow, bare land
- **Green Space** - garden land, arbour forest, shrub forest, park, natural meadow, artificial meadow
- **Urban Area** - industrial area, urban residential, rural residential, stadium, square, road, overpass, railway station, airport

The processed tiles are assigned to training, validation, and test sets in a standard 70/20/10 split, with assignment balanced across the splits during generation. Data augmentation is not applied during the preprocessing stage, and is instead applied to the tiles during training, which is discussed in Section 4.3.8. The results of this preprocessing can be seen in Figure 4.1.

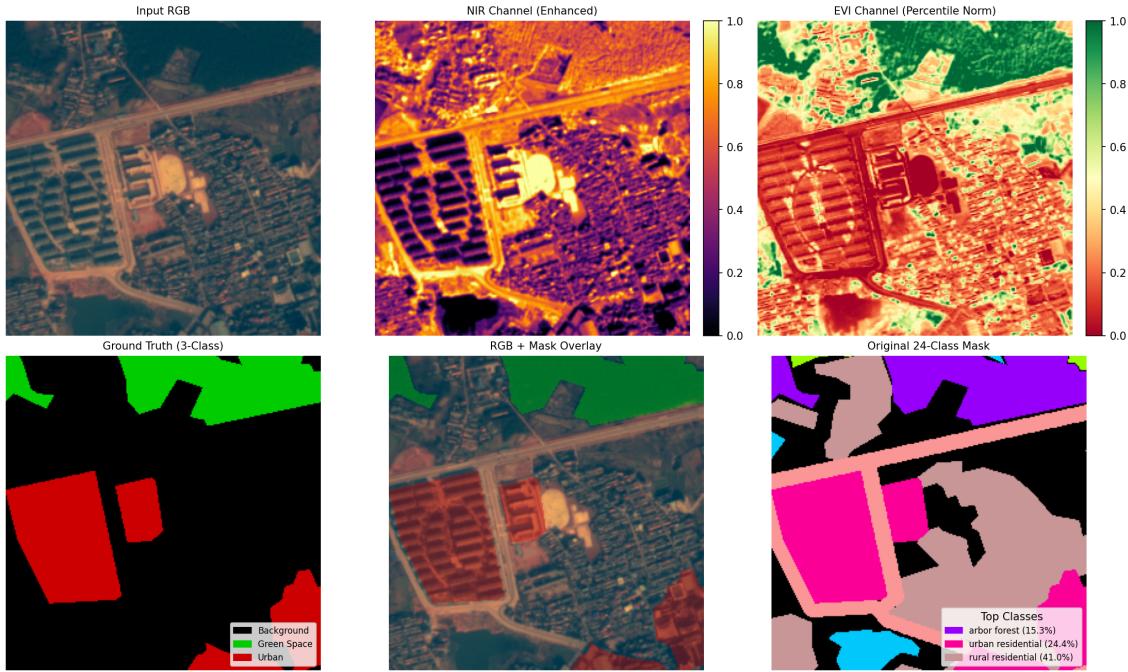


Figure 4.1: This shows a 2x3 grid of different visualisations of spectral channels that get used for training by the BSMs. Note, these tiles show human-readable versions of the channels used for training, i.e. RGB instead of BGR, and adjusted values for emphasis on boundaries, the actual tiles would be unedited raw data for the BSMs to use. From top to bottom and left to right, the first tile shows the RGB channels, the second tile shows the NIR channel, the third tile shows the EVI channel, the fourth tile shows our three class ground truth mask, the fifth channel shows the three class ground truth overlaid onto the RGB tile, and the last tile shows the original 24 class ground truth mask for the tile.

4.2 Segmentation Model

In this section we cover our architectural choices for the BSMs used. We discuss our choice to use DeepLabV3+ with a ResNet18 backbone, and we explain the loss function being used to mitigate the class imbalance present in the FBP dataset.

4.2.1 DeepLabV3+

Both BSMs use the DeepLabV3+ architecture [10], selected for its success in pixel-level image segmentation and remote sensing tasks [18, 19]. DeepLabV3+ uses a Atrous Spatial Pyramid Pooling (ASPP) module and lightweight decoder module, to help define segmentation boundaries. It works by upsampling high-level features using the ASPP module and combining them with low-level features extracted earlier in the encoder [10]. The low-level features are reduced in channel depth via a convolution layer, and then are concatenated with the upsampled features [10]. Additional specifics of this architecture can be seen in Figure 4.2

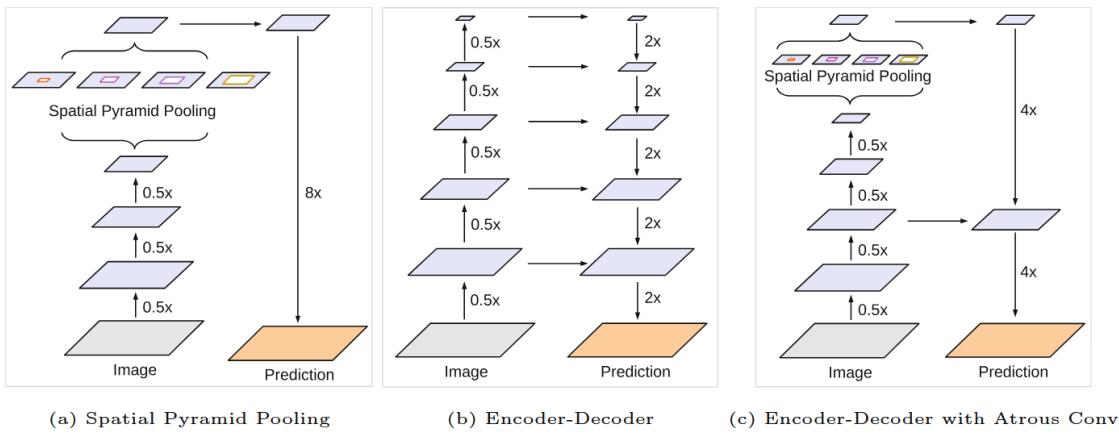


Figure 4.2: Architecture image taken from original DeepLabV3+ paper [10].

4.2.2 ResNet Backbone

The encoder portion of the DeepLabV3+ models is implemented using a ResNet18 backbone [20]. This backbone uses residual connections to improve the training of deep networks by mitigating the vanishing gradient problem [21]. ResNet backbones have seen use in other remote sensing projects, mainly for their balance between efficiency and depth [22]. This transfer learning approach is standard practice in deep learning and has shown consistent performance when applied to satellite imagery [23]. The decision to use transfer learning over a untrained U-Net is further justified in the final short discussion of this paper, in Chapter 7.

As the input data for this project includes spectral bands beyond the standard RGB channels, the first convolutional layer of the ResNet18 backbone has to be modified so that it can accept five input channels, instead of the default three [24], enabling the model to take full advantage of

the multispectral information available, which is necessary for allowing the model to distinguish between various land cover types [25, 26].

4.2.3 Class Imbalance

As previously mentioned in Chapter 3, the dataset is largely imbalanced towards background pixels, which is to be expected for a segmentation task that focuses on less frequent classes. To mitigate this, we employ the following weighted cross-entropy loss function [27], a common technique to counteract class imbalance in semantic segmentation [28, 29]. This is shown in Equation 4.2:

$$L_{WCE} = -\frac{1}{N} \sum_{i=1}^N [w_1 \cdot y_i \log(p_i) + w_0 \cdot (1 - y_i) \log(1 - p_i)] \quad (4.2)$$

where y_i represents a binary label of either the positive class (1) or negative class (0) for pixel i , and p_i is the model’s predicted probability for that pixel belonging to the positive class. The terms w_1 and w_0 counteract the class imbalance, as they are the weights assigned to the positive class and negative class respectively. In our active learning framework, they are calculated for each training batch based on the inverse class frequency [27] within the currently labelled training set D_L . This can be seen in Equation 4.3:

$$w_c = \frac{|D_L|}{|D_{L,c}|} \quad (4.3)$$

where c is either the positive class (1) or negative class (0), D_L is the total number of pixels in the current labelled training set, and $|D_{L,c}|$ is the number of pixels belonging to class c within the labelled training set. This weighting ensures that the model is penalised for errors on the less frequent target classes throughout the active learning process [27].

4.3 Active Learning Framework

Our AL framework presents a sequential decision-making problem which we tackle using two independent Deep Q-Network (DQN) agents [30] A_G and A_U . We choose a value-based algorithm that works to select informative tiles from the unlabelled pool D_U and add them to the shared labelled pool D_L , in order to guide the training process for the corresponding BSMs M_G and M_U .

4.3.1 State Representation

At the beginning of each AL cycle/epoch t , we construct a shared state vector S_t . This state summarises the system's current condition for both agents, and is seen below in Equation 4.4.

$$S_t = \begin{bmatrix} \text{IoU}_{G,t-1} \\ \text{IoU}_{U,t-1} \\ \text{Loss}_{G,t-1} \\ \text{Loss}_{U,t-1} \\ \mu(H_G(U_{\text{cand}})) \\ \sigma(H_G(U_{\text{cand}})) \\ \mu(H_U(U_{\text{cand}})) \\ \sigma(H_U(U_{\text{cand}})) \\ t \\ |U_t|/|U_0| \end{bmatrix} \quad (4.4)$$

where $\text{IoU}_{k,t-1}$ and $\text{Loss}_{k,t-1}$ are the last validation metrics for model k , which is included so that the model can observe the direct impact of the previous tile selection. $\mu(H_k(U_{\text{cand}}))$ and $\sigma(H_k(U_{\text{cand}}))$ represent the mean and standard deviation of model k 's predicted entropy across a candidate subset U_{cand} made up from the unlabelled pool. t and $|U_t|/|U_0|$ are the cycle progress and remaining unlabelled data budget that are added to give the model some temporal context about the current training state.

4.3.2 Tile Acquisition Strategy

Instead of selecting actions based on a standard exploration algorithm such as epsilon greedy, the agent's Q-values directly decide the exploration weights α_k used when acquiring tiles. The agent does this by using its estimated Q-value to balance between exploiting tiles with high uncertainty values, and exploring random tiles that may not have a high uncertainty value according to the current model. The result is that a larger Q-value results in larger alpha values α_k encouraging high entropy/uncertain tiles to be selected, and a low Q-value, which results in lower alpha values α_k which in turn encourage exploration.

This approach is beneficial as when the agent predicts a high Q-value for the current state S_t , it predicts that there will be an increase in performance/reward, so it will continue exploiting what it knows i.e. that selecting tiles the model is uncertain about is improving performance. However, if the agent predicts a low reward from the current state and therefore a low Q-value, then the result will be a low alpha value α_k , which will shift the strategy to more random tile selections. This exploration helps the system to gather more samples and discover regions of feature space where the model has made incorrect assumptions about a tile's uncertainty. This strategy doesn't force the agent to explore unnecessarily, and therefore only adds potentially uninformative tiles into the labelled pool D_L when exploration is necessary to further the model's uncertainty confidence.

This workflow cycle is outlined below:

1. The framework samples the candidate set U_{cand} made up of around 10% of the tiles from the unlabelled pool D_U
2. The average pixel-wise entropy ($H_k(j)$) is then computed and normalised for each candidate tile $j \in U_{\text{cand}}$ for both models $k \in \{G, U\}$
3. We then obtain the raw Q-values $Q_k(S_t)$ from each agent A_k from the state S_t
4. These Q-values are then normalised into exploration weights α_k based on the observed range $[Q_{\min,k}, Q_{\max,k}]$ for agent k :

$$\alpha_k = \text{clip}\left(\frac{Q_k(S_t) - Q_{\min,k}}{Q_{\max,k} - Q_{\min,k} + \epsilon}, 0, 1\right)$$

5. Then for each candidate tile j , we generate a random score $\rho_j \sim U(0, 1)$ and create a blended acquisition score: $\text{Score}_k(j) = \alpha_k(H_k(j)) + (1 - \alpha_k)\rho_j$
6. Finally, we select the top $\frac{B}{2}$ tiles for both models based on $\text{Score}_{\text{Green}}$ and $\text{Score}_{\text{Urban}}$ respectively, where B represents the total number of tiles we are adding to the labelled pool D_L . The union of these tile selections forms the batch I_{selected} acquired this cycle.

4.3.3 Reward Signal

The agents are rewarded based on the improvement in their corresponding model's performance. Following validation every few cycles (four cycles in our case), the reward $R_{k,t}$ for agent k is the change in the validation IoU score. This is displayed in Equation 4.5.

$$R_{k,t} = \text{clip}(\text{IoU}_{k,t_{\text{prev}}} - \text{IoU}_{k,t_{\text{curr}}}, R_{\min}, R_{\max}) \quad (4.5)$$

where t_{prev} and t_{curr} mark the previous validation cycle and the current validation cycle, and R_{\min}, R_{\max} defines the clipping range, in our case $[-0.1, 0.1]$, which we do to help stabilise the model's learning. This reward focuses the agents on learning a selection strategy that aims to increase the segmentation accuracy.

4.3.4 Network Architecture

The Q-value function $Q_k(S_t)$ for each agent k is approximated using a DQN, which is implemented as a Multi-Layer Perceptron (MLP). This MLP maps the 10-dimensional state vector S_t , shown in Section 4.3.1, through two fully-connected hidden layers of 64 and 32 neurons that use ReLU activation functions. This structure was selected for its suitability for processing a low dimensional state vector [30]. We chose this DQN architecture for approximating the Q-value function over a Q-table structure, due to the Q-table's known issue of the 'curse of dimensionality', which is the exponential computational cost associated with an increasing number of features in a dataset [31].

4.3.5 Experience Replay

To stabilise learning further, each agent employs an experience replay buffer [30, 32]. This buffer stores past experiences, allowing the agent to learn from a diverse set of stored transitions rather than just the most recent one. This breaks the temporal correlations in sequential experiences, which can lead to unstable training [32]. Our buffer stores up to 10,000 tuples, which is a standard buffer size for this type of environment. The stored tuples take the form $(S_t, R_{k,t})$, which associates the previous validation cycle state with the performance change after the current validation cycle.

4.3.6 Learning Update

Learning occurs after the validation cycle when new rewards are calculated. During the learning step, each agent samples a batch of size B of experience tuples T from the replay buffer. The network's weights are updated to minimise the Mean Squared Error (MSE) loss between the Q-value predicted by the local network $Q_{\text{local}}(S)$, and the actual reward R received for that experience [30]. This is shown in Equation 4.6.

$$\mathcal{L} = \frac{1}{B} \sum_T (R - Q_{\text{local}})^2 \quad (4.6)$$

This differs from other Q-learning projects, as our 'episodes' end after a single state-reward pair is stored, so we simplify the target to just be the immediate reward R , instead of the standard Bellman equation which includes a term for discounted future rewards [33]. The gradients are calculated based on this loss, and the network weights are updated using the Adam optimiser [34]. Additionally, gradient clipping is applied before the optimiser step to prevent excessively large updates and further stabilise training. While not common, this approach has been used in reinforcement learning environments where the focus is only on immediate outcomes from actions [35].

4.3.7 Target Network

To further stabilise the learning process, we introduce a target network Q_{target} with the same architecture as the local network Q_{local} [30]. This helps to prevent any rapid changes that can occur if the same network is used to predict the current Q-value and to estimate the target value [30]. The target network's weights θ' are updated after each learning step, to gradually track the local network's weights θ using a soft update rule. This is displayed in Equation 4.7.

$$\theta' \leftarrow \tau\theta + (1 - \tau)\theta' \quad (4.7)$$

where τ ($1e-3$ in our case, which is a standard value) controls the update rate.

4.3.8 Optimisation and Augmentation

Within each AL cycle, after updating the labelled pool D_L , both BSMs (M_G and M_U) are retrained on this pool for four epochs. The Adam optimiser used has an initial learning rate $1e - 4$ and an exponential decay of 0.99, which is applied after every cycle. Additionally, we apply data augmentation to the batches loaded from D_L during training to increase generalisability and decrease any potential over-fitting. These augmentations include some light noise and Gaussian blurring, random horizontal and vertical flips, and multiple 90 degree image rotations i.e. 90, 180, 270 degree rotations.

4.4 UGS Definition from Expert Labelling

While the BSMs learn to identify Green Space and Urban Area pixels based on the three class masks ground truths from the FBP dataset, the precise definition of an UGS ground truth is not immediately available. Therefore, to create a ground truth for the final UGS classification and overall evaluation, we perform manual labelling at the tile-level to gain 'expert knowledge' of whether a tile contains an UGS or not. This involves reviewing the RGB and mask channels for a tile and making a binary decision of if the tile is an instance of an UGS or not. While many cases are straight forward, some tiles are subjective and depend on factors such as urban area connectivity and ratio.

This labelling process resulted in a set of binary labels assigned to tiles from the labelled training dataset D_L , and becomes the ground truth for training the model covered in section 4.5.

4.5 UGS Tile Classifier Model

To use the expert tile-level UGS definitions for evaluating the BSM outputs, we train a Tile Classifier Model (TCM) (M_{TC}) that learns to predict the expert UGS label, 0 (not UGS) or 1 (UGS), for a given tile based on the segmentation predictions generated by the trained BSMs.

Input for the TCM consists of the prediction masks produced by the BSMs. For each tile j in the labelled training dataset D_L , inference is run using the trained M_G and M_U to produce the probability maps/binary masks $P_{Green}[j]$ and $P_{Urban}[j]$. These channels are then stacked to create a two-channel input shape of $2 \times 512 \times 512$ for each tile. The TCM has a CNN architecture with a ShuffleNet-V2-x0.5 backbone, chosen for its speed and effectiveness when handling simple tasks such as this binary tile classification [36]. The network outputs a single logit per tile, indicating how likely a tile is to contain an UGS according to the expert labels. The loss function and optimiser used is a Binary Cross-Entropy with Logits loss function from the standard PyTorch library¹, and the Adam optimiser.

¹https://pytorch.org/docs/stable/generated/torch.nn.functional.binary_cross_entropy_with_logits.html

4.6 Evaluation

The final evaluation creates quantitative and qualitative metrics to show the performance of the complete UGS segmentation system on the test set (D_T).

4.6.1 Inference

Before evaluating begins, inference is run for each tile in the test set:

1. We run inference with the trained BSMs to obtain prediction masks $P_{Green}[j]$ and $P_{Urban}[j]$.
2. Then the predictions are stacked into the two-channel format $2 \times 512 \times 512$.
3. We then perform inference with the trained TCM and stacked predictions as input to get a tile-level prediction $P_{Tile}[j] \in \{0, 1\}$.

4.6.2 UGS Mask Generation

To evaluate the system against the expert definition of an UGS, both the final predicated segmentation mask P_{UGS} and the corresponding ground truth mask G_{UGS} are constructed through a binary tile-level evaluation process that coordinates the system's assessment with the expert tile-level annotation.

The predicted UGS mask is generated based on the TCM output. For each test tile, the predicted green space pixels are only included in the mask if the tile is classified as an UGS, otherwise an empty mask of all zero values is used:

$$P_{UGS}[j] = \begin{cases} P_{Green}[j] & \text{if } P_{Tile}[j] = 1 \\ 0 & \text{if } P_{Tile}[j] = 0 \end{cases} \quad (4.8)$$

This means only pixels predicted as green space, that are also classified as UGS by the TCM, contribute to the final predication map evaluated.

The ground truth UGS mask used for comparison is constructed similarly by using the expert tile-level labels. If the expert labelled test tile contains UGS, then the corresponding area contains the actual green space pixels from the three class mask ground truth mask for that tile. This method ensures that the final evaluation metrics compare the system's ability to segment green space pixels specifically within the tiles identified as UGS instances by the expert labels. The results of this evaluation method can be seen in Chapter 6.

4.6.3 Metrics

Final system performance is judged using standard segmentation metrics, comparing the predicted P_{UGS} against the constructed ground truth G_{UGS} . The metrics we look at include Intersection-

4. Methodology

over-Union (IoU), Precision, Recall, and F1-score, treating UGS (class 1) as the positive class. Descriptions for these metrics are described below:

- **Intersection-over-Union (IoU):** Also known as the Jaccard Index, this measures the overlap between the predicted and ground truth regions for the positive class.

$$\text{IoU} = \frac{\text{TP}}{\text{TP} + \text{FP} + \text{FN}} \quad (4.9)$$

- **Precision:** Measures what proportion of pixels predicted as UGS actually are UGS (positive predictions).

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}} \quad (4.10)$$

- **Recall:** Measures what proportion of actual UGS pixels were correctly identified.

$$\text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}} \quad (4.11)$$

- **F1-Score:** The mean value of the precision and recall metrics, providing a single well rounded score.

$$\text{F1} = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} \quad (4.12)$$

4.7 Active Learning Algorithm

To consolidate the overall workflow, here is a pseudo-code algorithm outlining the important steps to our AL framework.

Algorithm 1: Active Learning with DQN Agents for UGS Segmentation

Data: Initial Labelled D_L , Unlabelled Pool D_U , Validation D_V , Test D_T

Result: Final Overall Metrics on D_T

- 1 Models:** M_G , (Green space), M_U (Urban area), M_{TC} (Tile Classifier)
- 2 Agents:** A_G, A_U
- 3 Hyper-parameters:** $N_{cycles}, N_{epochs}, B$ (Batch size), N_{cand} (no of candidates), R_{clip} (Clip range)
- 4 Variable Legend:** S_t (State at time t), E_G, E_U (Model entropy scores), R (Random scores for exploration), I_{cand} (Candidates sample indices)
- 5** Train M_G, M_U on D_L initially
- 6** $Metrics_{prev} \leftarrow \text{Validate}(M_G, M_U, D_V)$ # Metrics are losses and IoUs
- 7** **for** $cycle \leftarrow 1$ **to** N_{cycles} **do**
- 8** $S_t, E_G, E_U, R, I_{cand} \leftarrow \text{CalcStateAndScore}(M_G, M_U, D_U, Metrics_{prev}, cycle, N_{cycles}, N_{cand})$
- 9** $Q_G \leftarrow A_G.\text{get_q_value}(S_t)$
- 10** $Q_U \leftarrow A_U.\text{get_q_value}(S_t)$
- 11** $\alpha_G \leftarrow \text{Clip}(\text{Normalise}(Q_{G,\text{raw}}, A_G.\text{range}), 0, 1)$ # Get alpha values
- 12** $\alpha_U \leftarrow \text{Clip}(\text{Normalise}(Q_{U,\text{raw}}, A_U.\text{range}), 0, 1)$
- 13** $C_G \leftarrow \alpha_G E_G + (1 - \alpha_G)R$ # Calculate candidate scores
- 14** $C_U \leftarrow \alpha_U E_U + (1 - \alpha_U)R$
- 15** $I_{selected} \leftarrow \text{SelectTopIndices}(C_G, C_U, B, I_{cand})$
- 16** $D_L, D_U \leftarrow \text{UpdatePools}(D_L, D_U, I_{selected})$
- 17** **for** $epoch \leftarrow 1$ **to** N_{epochs} **do**
- 18** | Fine-tune $M_G(\theta_G)$ and $M_U(\theta_U)$ on D_L # θ is weights and biases
- 19** **end**
- 20** $Metrics_{curr} \leftarrow \text{Validate}(M_G, M_U, D_V)$
- 21** $\Delta IoU_G, \Delta IoU_U \leftarrow \text{CalculateDeltaIoU}(Metrics_{curr}, Metrics_{prev})$
- 22** $r_G \leftarrow \text{Clip}(\Delta IoU_G, R_{clip,min}, R_{clip,max})$ # Get clipped reward values
- 23** $r_U \leftarrow \text{Clip}(\Delta IoU_U, R_{clip,min}, R_{clip,max})$
- 24** $A_G.\text{step}(State_t, r_G)$
- 25** $A_U.\text{step}(State_t, r_U)$
- 26** $Metrics_{prev} \leftarrow Metrics_{curr}$
- 27** **end**
- 28** Calculate overall metrics using M_G, M_U, M_{TC} on D_T

Chapter 5

Experiments

This section details the experimental setup, baseline methods, evaluation protocol, and metrics used to assess the performance and efficiency of the proposed AL framework for UGS segmentation.

5.1 Hardware Specifications

All data preprocessing and model training sessions were performed on an NVIDIA RTX 3070ti GPU with 8GB of VRAM and a 12th Gen Intel i7-12700F with 32GB of RAM. This resulted in a training times of approximately 12 hours for 100 AL cycles. This training time, combined with a short project life cycle, made hyper-parameter tuning especially challenging. As a result, we had to carefully decide which parameters to test and settle on a final reward structure for the AL agent early in the process, to allow any time for tuning. The implications of these constraints are discussed further in Chapter 7.

5.2 Hyper Parameters

Here we present the results of some of the hyper-parameters that were tested, giving evidence as to why exact values were selected. In these tests, we look at the IoU of the green space segmentation model, as the urban area model tends to perform comparatively, making it unnecessary to additionally document.

5.2.1 Segmentation Learning Rate

We tested learning rates an order of magnitude above and below the commonly used default of $1e-4$, and evaluated their impact on the segmentation performance. The results of this are summarised in Figure 5.1, and show that a learning rate of $1e-4$ achieved the highest IoU score, making it the clear choice for our model configuration.

5.2.2 Tile Size

We tested two standard tile sizes for satellite segmentation tasks, the results of which can be seen in Figure 5.2. This displays an obvious performance increase when training with a tile size of 512. One explanation for this could be that segmentation tasks often heavily rely on having sufficient contextual information. Understanding if a patch of green is a small park surrounded by buildings, part of a large forest, or agricultural land, requires being aware of the surroundings area. Larger tiles would allow the model to see more of these spatial relationships, reducing ambiguity and improving classification accuracy, especially near boundaries.

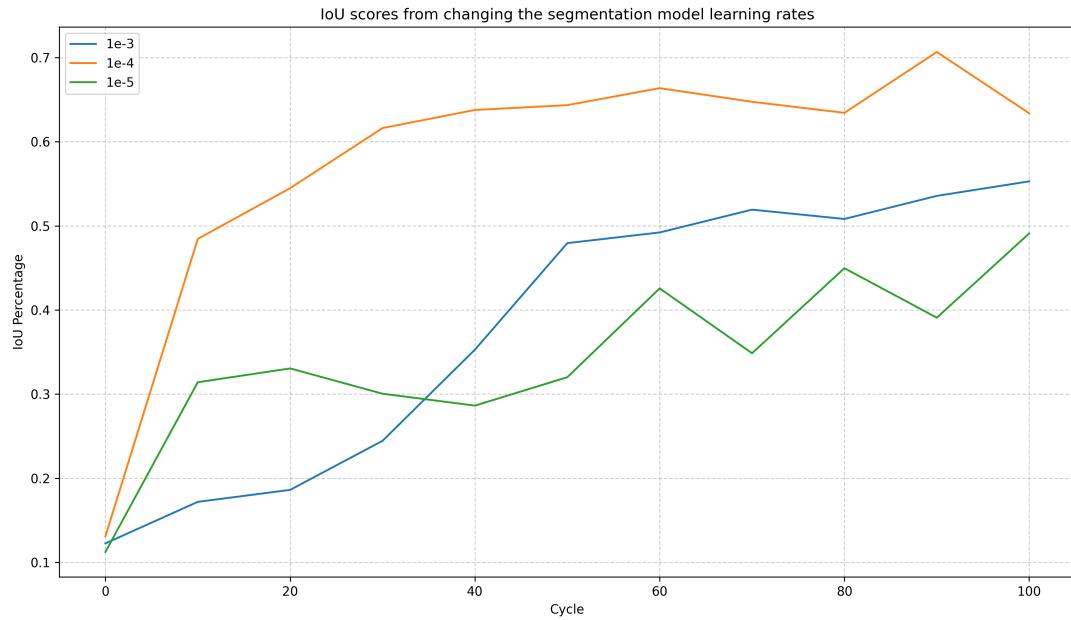


Figure 5.1: IoU scores from changing the segmentation model learning rate: $1e-3$, $1e-4$, and $1e-5$. The top line shows a learning rate of $1e-4$ consistently produced the highest performance, supporting its selection for the final model configuration.

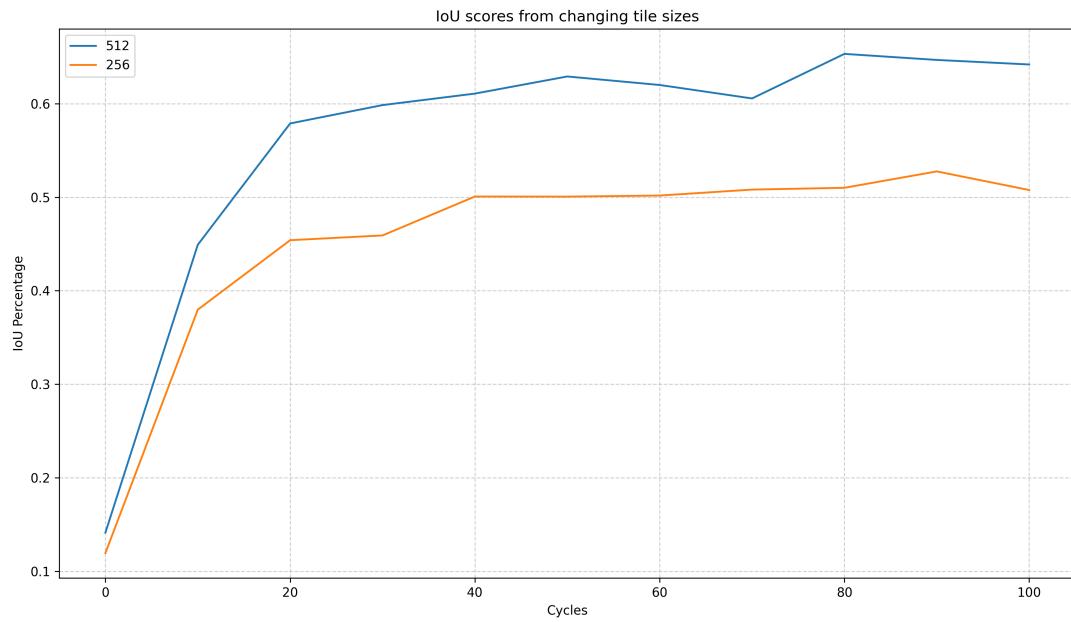


Figure 5.2: Comparison of segmentation performance using different tile sizes. This shows that the model trained with a tile size of 512 achieves a higher IoU than the model that was trained on images with a tile size of 256.

5.3 ResNet Depth

To evaluate the effect of backbone depth on segmentation performance, we tested three common ResNet backbones: ResNet50, ResNet34, and ResNet18. Each model was trained and validated under the same configuration to ensure a fair comparison. These metrics were obtained by evaluating each model on the test dataset, providing a fair and consistent benchmark. As shown in Figure 5.3, shallower backbones significantly outperformed the deeper ResNet50, offering both improved accuracy and reduced computational cost across all evaluated metrics. Whereas ResNet50 showed signs of over-fitting to the training data with considerably lower evaluation metrics.

ResNet18 achieved the highest scores for both Green Space and Urban segmentation, and therefore became the backbone used in the final segmentation model configuration. Its strong generalisation ability, coupled with lower computational requirements, made it well-suited for the task of UGS mapping. Additionally, any marginal gains that the ResNet34 backbone could potentially produce on subsequent training sessions would not be significant enough to justify the increased model complexity. Exact metric values can be seen in the relevant Appendix table B.

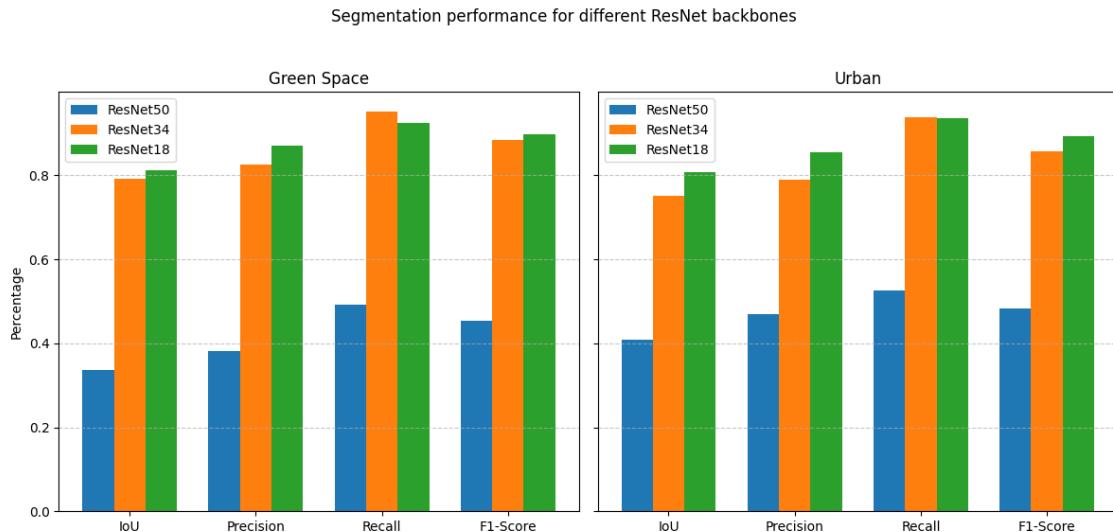


Figure 5.3: Performance comparison of DeepLabV3+ models with ResNet50, ResNet34, and ResNet18 encoders across segmentation metrics for Green Space and Urban classes. Metrics were computed on the test dataset, revealing that shallower backbones not only generalised better but also avoided over-fitting, particularly in contrast to the deeper ResNet50 model.

Chapter 6

Results

Here we present an evaluation of the proposed UGS segmentation framework. Performance is assessed using both quantitative and qualitative metrics gained from evaluating the framework’s performance against a test set constructed from the FBP dataset. The quantitative evaluation relies on standard segmentation metrics previously discussed in Section 4.6.3, and the qualitative assessment provides a more visually appealing perspective into the segmentation outputs, giving a unique outlook as to where the model succeeded and where there may have been challenges.

Importantly, we saw a promising level of data efficiency from the AL approach. All performance results were obtained using a labelled training set of only 2,469 tiles, which is a massive reduction from the FBP’s dataset size of approximately 28,000 tiles. Meaning the presented segmentation accuracy was achieved through training on only 8.8% of the total dataset. This highlights the framework’s effectiveness in significantly reducing the manual annotation effort typically required for training segmentation models.

6.1 Quantitative Metrics

Below are two bar charts that show numerical evaluations (see Figures 6.1 and 6.2). The first bar chart shows a baseline model that was trained by selecting tiles to be added to the labelled training pool, using a random tile selection strategy. Along side this, we show the performance results of the most promising model that selected tiles using our AL framework.

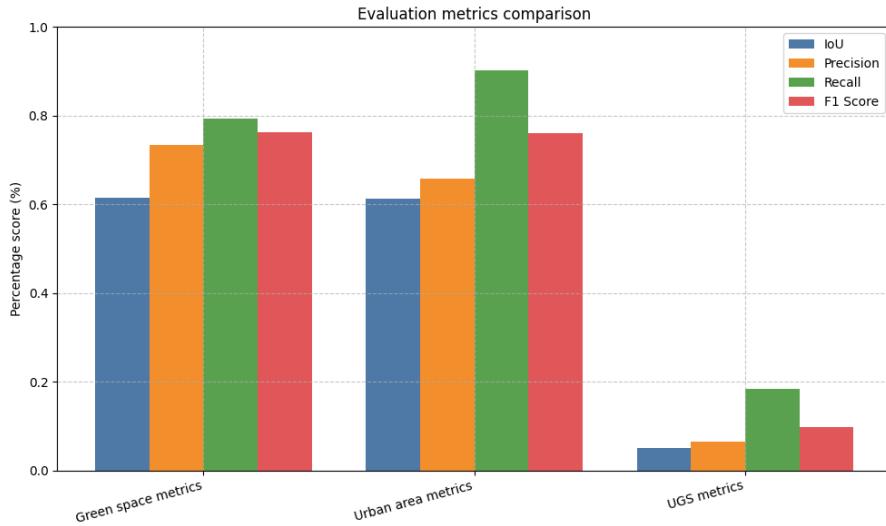


Figure 6.1: This bar chart displays the evaluation metrics corresponding to results from the BSMs that were trained using randomly selected tiles each cycle of training. It shows a noticeable drop in performance from both the green space and urban area models, and especially poor performance when evaluating the UGS metric with the TCM. The UGS metric results are likely poor due to the compounding error from the initial green space and urban area model's performances.

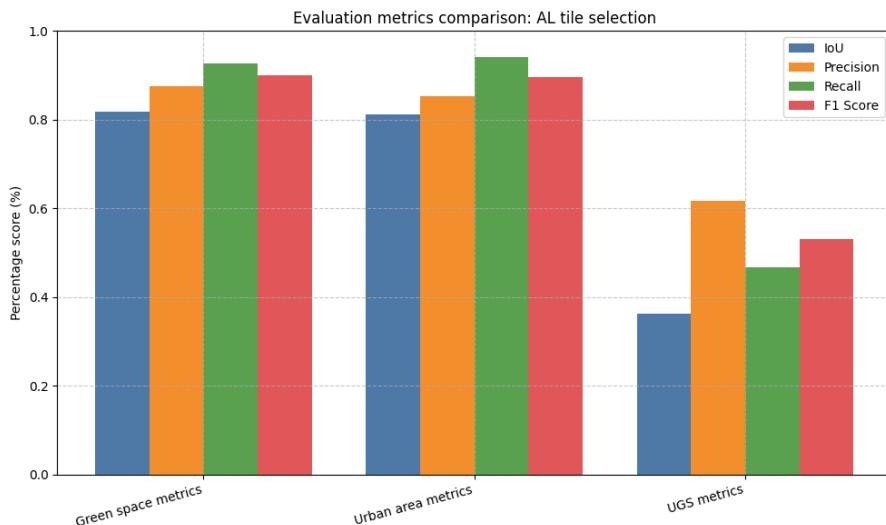


Figure 6.2: This bar chart displays the evaluation metrics corresponding to results from the BSMs that were trained using the AL framework to select tiles each cycle of training. It shows improvement compared to the baseline results in Figure 6.1, and considerable improvement in the UGS metrics when compared to the baseline results.

The following graphic is a direct comparison between our baseline and AL framework results that were just shown, but in the form of a radar graph (see Figure 6.3). This lets us see a clear representation of how the AL framework directly improves upon a basic random tile selection method.

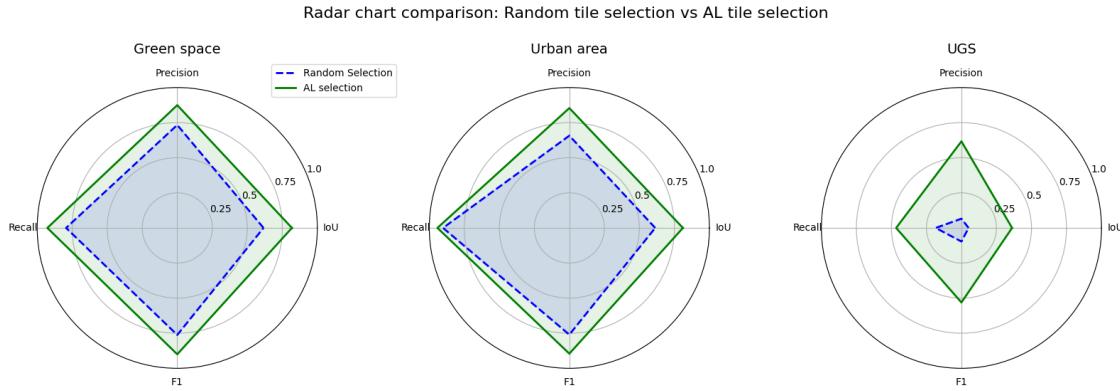


Figure 6.3: This displays an alternative visualisation of the baseline and AL framework metrics, in the form of a radar graph.

The next two line graphs display the alpha and Q-values that were measured throughout the AL framework's training session (see Figures 6.4 and 6.5).

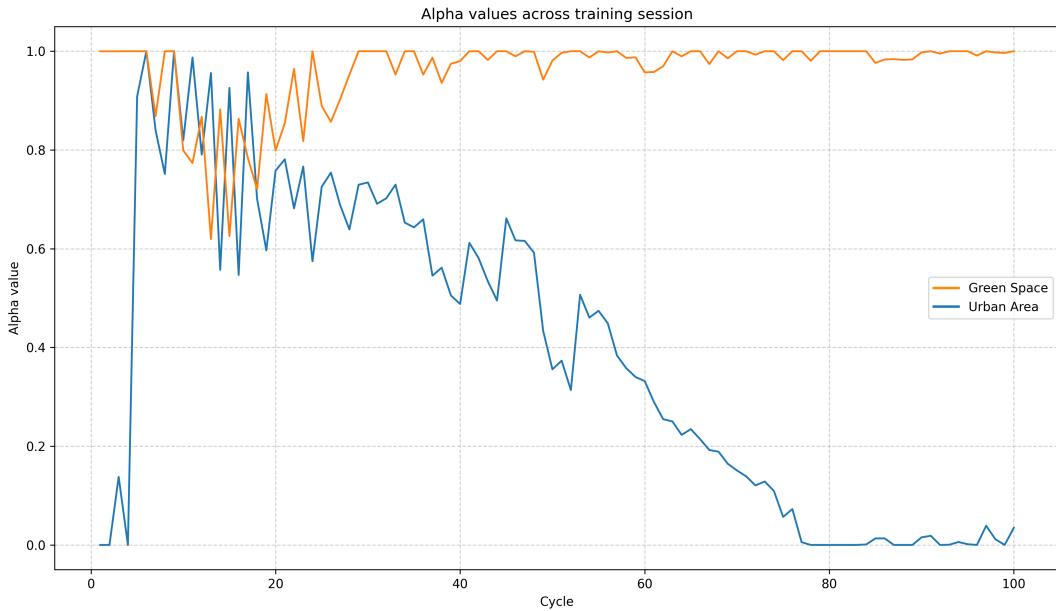


Figure 6.4: This directly demonstrates the exploration strategies that were used by the green space and urban area agents. We can see from the line graph that the green space agent learned that mostly exploiting the uncertainty values to make informed tile selections, had a high chance of producing IoU improvements. On the other hand, the urban area agent learned that initially selecting tiles using uncertainty knowledge led to some IoU improvements, but quickly switched to a more balanced strategy of both exploitation and exploration, and eventually ending with complete exploration, likely due to a plateau in potential IoU gains.

6. Results

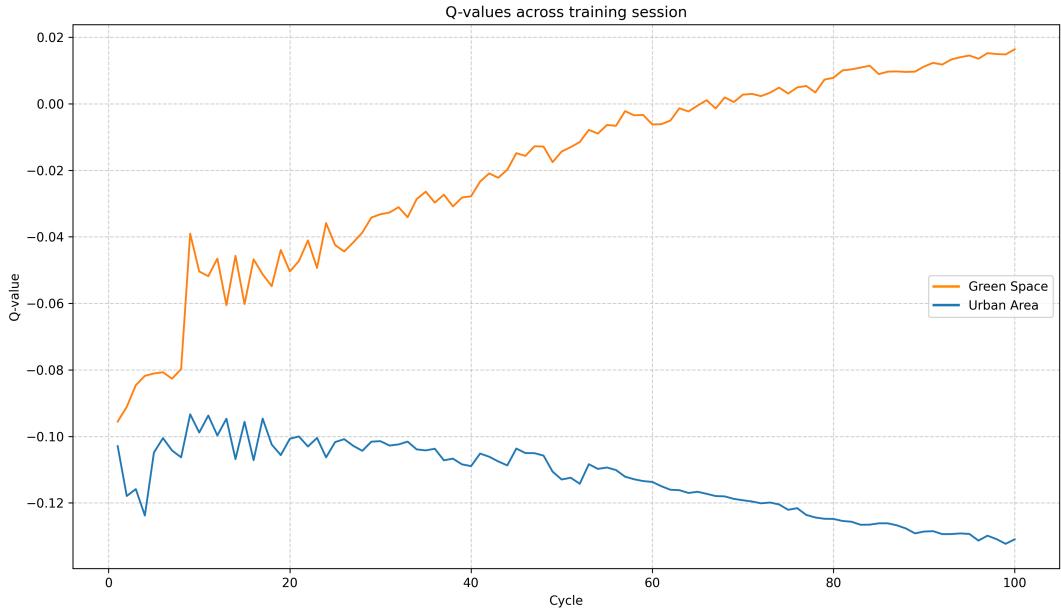


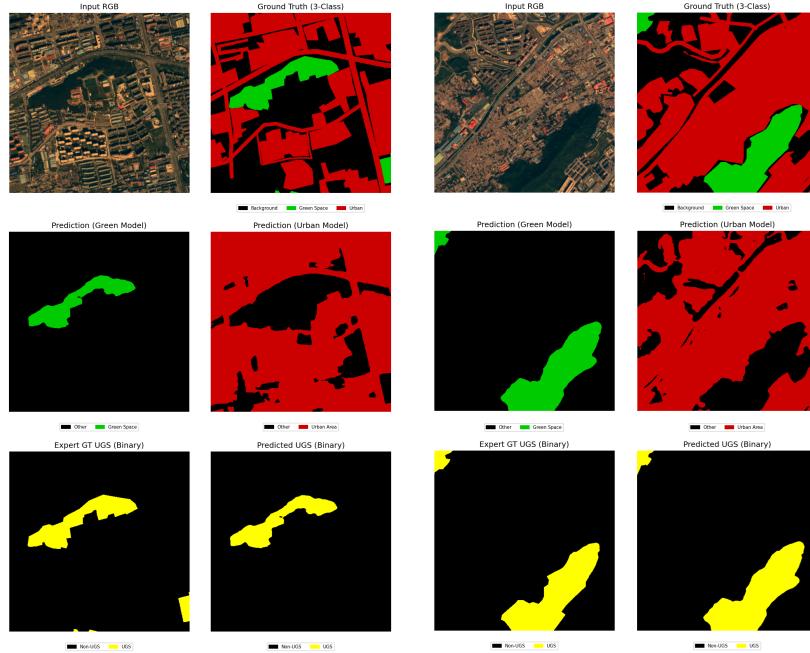
Figure 6.5: This line graph shows the Q-values estimated during training, which are used to derive the corresponding alpha values that are responsible for the agent’s exploration strategy. Similarly to the alpha value line graph in Figure 6.4, we can see that the green space agent discovers a positive association between exploiting uncertainty values and increased IoU performance. However, the urban area agent does not sustain this same association. Instead, its Q-value fluctuates and eventually decreases, suggesting that there are diminishing expected rewards for exploitation, which pushes the agent to explore, which is consistent with Figure 6.4.

6.2 Qualitative Metrics

Our second evaluation shows a number of tile predictions from the green space and urban area BSMs. The first two images show the results of tile predictions that were highly successful in their pixel-level UGS classification. The second two images show the results of predictions that over-predicted the UGS areas in the image, showing a potential current limitation of the BSMs abilities. However, it’s important to still keep in mind that the ground truths in these images have been derived from the expert manual labelling process at the tile-level, as previously discussed in Section 4.4, so must be viewed critically as a useful but still imperfect comparison.

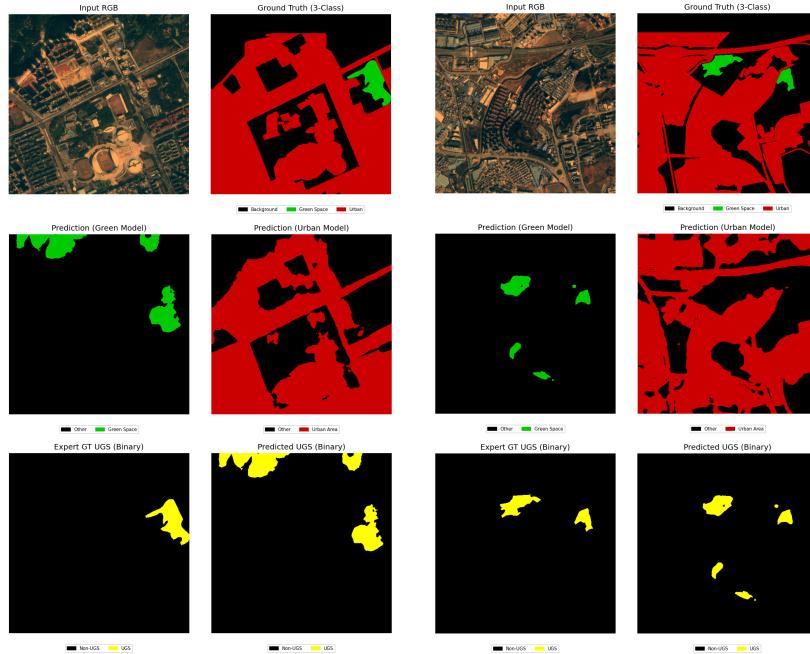
Here is a brief description of the format for these tile prediction visualisation examples:

- The top row of the example images shows both the Input RGB and the three class ground truth mask.
- The middle row shows the inferences made from the green space and urban area BSMs.
- The bottom row shows the expert UGS ground truth on the left against the predicted UGS ground truth on the right.



A. Mostly correct UGS prediction - example 1

B. Mostly correct UGS prediction - example 2



C. Over-predicted UGS - example 1

D. Over-predicted UGS - example 2

Figure 6.6: This shows four example tile predictions. The top two images demonstrate mostly correct pixel-level UGS classification, i.e. the BSMs predicted most true positive pixels on the expert ground truth, and kept false negatives to a minimum. The bottom two images demonstrate over-predicted pixel-level UGS classification, i.e. the BSMs predicted areas on false positive pixels that are not considered UGS on our expert ground truth.

6. Results

This final visualisation shows a composite of multiple tile predictions formed into a full satellite image. At a glance this demonstrates a promising overall performance from the project. However, on closer inspection, there are a number of false positive and false negative pixel classifications, resulting in many UGS areas being missed or falsely labelled.

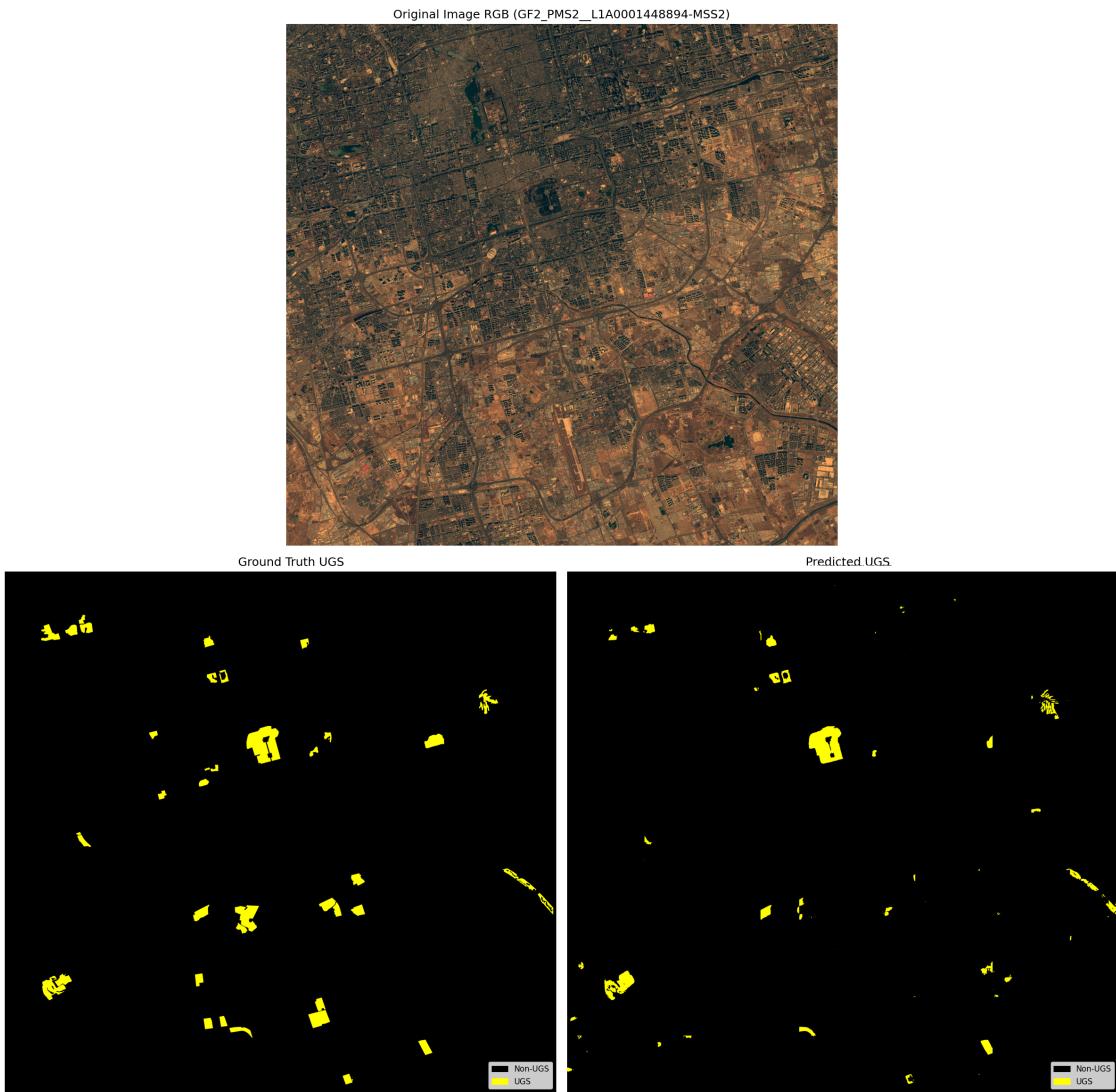


Figure 6.7: This figure displays a full satellite image and its corresponding ground truth and predictions from the BSMs. The image is a composite of all the evaluated tiles belonging to the original satellite image. This displays some promising results from the framework, and also problematic misclassification issues.

Chapter 7

Discussion

While the project demonstrates a successful implementation of an AL framework for UGS segmentation, there are several avenues for future work that are worth while discussing.

While the agents demonstrated noticeable adaptive behaviour, there remains a large scope for continued experimentation. For a thorough venture into improving the current AL setup, the state representation should be explored, as including different or richer features to describe the unlabelled pool could improve the agents' decision making ability. Similarly, investigating other reward signals, potentially in the form of label costs or other long-term performance indicators, could lead to improved AL performance and efficiency.

The current exploration strategy is controlled by the normalised Q-value, which represents just one of the many possible reward mechanisms that could be employed. Initial experimentation of an epsilon greedy strategy presented an interesting issue, where by having the agent explore for a non-trivial amount of time during training, in order to gain an understanding of what makes a tile informative, resulted in many tiles being added to the labelled pool that were inherently uninformative, a discovery that prompted the current exploration strategy. However, further experimentation into other RL exploration techniques or even a revisit to an epsilon greedy strategy would be highly beneficial to ensure any future improvements.

The most obvious potential criticism for this project is the hyper-parameter tuning and experimentation for the DQN agents, which was greatly restricted by a combination of the project's short time frame, and that each training run required up to around 12 hours of computation time on the available hardware. This made optimisation strategies such as grid searches mostly impractical but would be a clear way to further the AL agent's performance. It should also be noted that we originally proposed training a U-Net from scratch for the segmentation models, but quickly switched to transfer learning to make use of a pre-trained backbone. This had the benefit of being faster and much less complicated to train the mod, which was an important factor given the projects short lifecycle.

Lastly, while the results of some similar papers was briefly discussed in the related works Chapter 2, a more in-depth comparison to other similar projects could highlight other gaps in the

7. Discussion

project, and clearly show areas that require improvement. Along with the previously mentioned immediate areas for improvement, this would be an obvious next step for any future work on the project.

Chapter 8

Conclusion

This research implements and evaluates a framework for Urban Green Space (UGS) segmentation in satellite imagery, directly addressing the challenges of annotation costs. We introduce binary segmentation models that are built on the architecture DeepLabV3+ and use a ResNet18 backbone. These models leverage Active Learning (AL) agents built with a Deep Q-network (DQN) architecture, and are evaluated using a Tile Classifier Model (TCM) designed for classifying UGSs at the tile-level.

We demonstrated the feasibility of this approach, showing that the AL agents learned different adaptive strategies for balancing uncertainty sampling (exploitation) and random tile selection (exploration). The framework incorporates expert tile-level labels through the TCM, enabling the final UGS mapping to reflect an expert labelled definition. While the final quantitative performance against the expert UGS definition showed a potential issue with compounding errors resulting from the BSMs predictions being used in combination with the TCM to create a final UGS prediction, the qualitative results showed the system was able to combine pixel-level segmentation with tile-level classification on a basic level. Overall, this work presents a starting point into the exploration of an alternative methodology for label efficient UGS mapping using AL.

Bibliography

- [1] P. Matos, J. Vieira, B. Rocha, C. Branquinho, and P. Pinho, “Modeling the provision of air-quality regulation ecosystem service provided by urban green spaces using lichens as ecological indicators,” *Science of The Total Environment*, vol. 665, pp. 521–530, 2019. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S0048969719305029>
- [2] N. Kabisch, *The Influence of Socio-economic and Socio-demographic Factors in the Association Between Urban Green Space and Health.* Cham: Springer International Publishing, 2019, pp. 91–119. [Online]. Available: https://doi.org/10.1007/978-3-030-02318-8_5
- [3] M. Nastran, M. Kobal, and K. Eler, “Urban heat islands in relation to green land use in european cities,” *Urban Forestry & Urban Greening*, vol. 37, pp. 33–41, 2019, green Infrastructures: Nature Based Solutions for sustainable and resilient cities. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S1618866717304806>
- [4] X.-Y. Tong, G.-S. Xia, and X. X. Zhu, “Enabling country-scale land cover mapping with meter-resolution satellite imagery,” *ISPRS Journal of Photogrammetry and Remote Sensing*, vol. 196, pp. 178–196, 2023. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S0924271622003264>
- [5] R. E. Huerta, F. D. Yépez, D. F. Lozano-García, V. H. Guerra Cobián, A. L. Ferriño Fierro, H. de León Gómez, R. A. Cavazos González, and A. Vargas-Martínez, “Mapping urban green spaces at the metropolitan level using very high resolution satellite imagery and deep learning techniques for semantic segmentation,” *Remote Sensing*, vol. 13, no. 11, 2021. [Online]. Available: <https://www.mdpi.com/2072-4292/13/11/2031>
- [6] M. J. Rad and C. Armenakis, “Active reinforcement learning for the semantic segmentation of urban images,” *Canadian Journal of Remote Sensing*, vol. 50, no. 1, p. 2374788, 2024. [Online]. Available: <https://doi.org/10.1080/07038992.2024.2374788>
- [7] J. Tupayachi, M. M. Ferguson, and X. Li, “A simulation-based real-time deep reinforcement learning approach for fighting wildfires,” in *2024 Annual Modeling and Simulation Conference (ANNSIM)*, 2024, pp. 1–12.

- [8] A. Casanova, P. O. Pinheiro, N. Rostamzadeh, and C. J. Pal, “Reinforced active learning for image segmentation,” in *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. OpenReview.net, 2020. [Online]. Available: <https://openreview.net/forum?id=SkgC6TNFvr>
- [9] L. T. Ramos and A. D. Sappa, “Leveraging u-net and selective feature extraction for land cover classification using remote sensing imagery,” *Scientific Reports*, vol. 15, no. 1, p. 784, Jan. 2025.
- [10] L. Chen, Y. Zhu, G. Papandreou, F. Schroff, and H. Adam, “Encoder-decoder with atrous separable convolution for semantic image segmentation,” in *Computer Vision - ECCV 2018 - 15th European Conference, Munich, Germany, September 8-14, 2018, Proceedings, Part VII*, ser. Lecture Notes in Computer Science, V. Ferrari, M. Hebert, C. Sminchisescu, and Y. Weiss, Eds., vol. 11211. Springer, 2018, pp. 833–851. [Online]. Available: https://doi.org/10.1007/978-3-030-01234-2_49
- [11] M. Hu, J. Zhang, L. Matkovic, T. Liu, and X. Yang, “Reinforcement learning in medical image analysis: Concepts, applications, challenges, and future directions,” *CoRR*, vol. abs/2206.14302, 2022. [Online]. Available: <https://doi.org/10.48550/arXiv.2206.14302>
- [12] B. Srikanth, A. Tabassum, S. Allu, R. Kannan, and N. Muralidhar, “Reinforcement learning as a parsimonious alternative to prediction cascades: A case study on image segmentation,” in *Thirty-Eighth AAAI Conference on Artificial Intelligence, AAAI 2024, Thirty-Sixth Conference on Innovative Applications of Artificial Intelligence, IAAI 2024, Fourteenth Symposium on Educational Advances in Artificial Intelligence, EAAI 2024, February 20-27, 2024, Vancouver, Canada*, M. J. Wooldridge, J. G. Dy, and S. Natarajan, Eds. AAAI Press, 2024, pp. 15 066–15 074. [Online]. Available: <https://doi.org/10.1609/aaai.v38i13.29428>
- [13] N. Zeng, H. Li, Z. Wang, W. Liu, S. Liu, F. E. Alsaadi, and X. Liu, “Deep-reinforcement-learning-based images segmentation for quantitative analysis of gold immunochromatographic strip,” *Neurocomputing*, vol. 425, pp. 173–180, 2021. [Online]. Available: <https://doi.org/10.1016/j.neucom.2020.04.001>
- [14] G. Cheng and J. Han, “A survey on object detection in optical remote sensing images,” *ISPRS Journal of Photogrammetry and Remote Sensing*, vol. 117, pp. 11–28, 2016. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S0924271616300144>
- [15] R. Mathieu, J. Aryal, and A. K. Chong, “Object-based classification of ikonos imagery for mapping large-scale vegetation communities in urban areas,” *Sensors*, vol. 7, no. 11, pp. 2860–2880, 2007. [Online]. Available: <https://www.mdpi.com/1424-8220/7/11/2860>

- [16] M. Buda, A. Maki, and M. A. Mazurowski, “A systematic study of the class imbalance problem in convolutional neural networks,” *Neural Networks*, vol. 106, pp. 249–259, 2018. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S0893608018302107>
- [17] Kamel Didan and Armando Barreto Munoz, *MODIS Vegetation Index User’s Guide (MOD13 Series)*, https://lpdaac.usgs.gov/documents/621/MOD13_User_Guide_V61.pdf, LP DAAC, USGS, 2019, collection 6.1. Vegetation Index and Phenology Lab, The University of Arizona. Last Update September, 2019. [Online]. Available: https://lpdaac.usgs.gov/documents/621/MOD13_User_Guide_V61.pdf
- [18] E. Maggiori, Y. Tarabalka, G. Charpiat, and P. Alliez, “High-resolution aerial image labeling with convolutional neural networks,” *IEEE Transactions on Geoscience and Remote Sensing*, vol. 55, no. 12, pp. 7092–7103, 2017.
- [19] N. Audebert, B. Le Saux, and S. Lefèvre, “Beyond rgb: Very high resolution urban remote sensing with multimodal deep networks,” *ISPRS Journal of Photogrammetry and Remote Sensing*, vol. 140, pp. 20–32, 2018, geospatial Computer Vision. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S0924271617301818>
- [20] J. He, J. Chen, X. He, J. Gao, L. Li, L. Deng, and M. Ostendorf, “Deep reinforcement learning with a natural language action space,” in *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics, ACL 2016, August 7-12, 2016, Berlin, Germany, Volume 1: Long Papers*. The Association for Computer Linguistics, 2016. [Online]. Available: <https://doi.org/10.18653/v1/p16-1153>
- [21] K. He, X. Zhang, S. Ren, and J. Sun, “Deep residual learning for image recognition,” *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 770–778, 2015. [Online]. Available: <https://api.semanticscholar.org/CorpusID:206594692>
- [22] N. Kussul, M. Lavreniuk, S. Skakun, and A. Shelestov, “Deep learning classification of land cover and crop types using remote sensing data,” *IEEE Geoscience and Remote Sensing Letters*, vol. 14, no. 5, pp. 778–782, 2017.
- [23] O. A. B. Penatti, K. Nogueira, and J. A. dos Santos, “Do deep features generalize from everyday objects to remote sensing and aerial scenes domains?” in *2015 IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, 2015, pp. 44–51.
- [24] K. Makantasis, K. Karantzalos, A. Doulamis, and N. Doulamis, “Deep supervised learning for hyperspectral data classification through convolutional neural networks,” in *2015 IEEE International Geoscience and Remote Sensing Symposium (IGARSS)*, 2015, pp. 4959–4962.

- [25] X. X. Zhu, D. Tuia, L. Mou, G.-S. Xia, L. Zhang, F. Xu, and F. Fraundorfer, “Deep learning in remote sensing: A comprehensive review and list of resources,” *IEEE Geoscience and Remote Sensing Magazine*, vol. 5, no. 4, pp. 8–36, 2017.
- [26] G. Vivone, “Multispectral and hyperspectral image fusion in remote sensing: A survey,” *Information Fusion*, vol. 89, pp. 405–417, 2023. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S1566253522001312>
- [27] M. Cabezas and Y. Diez, “An analysis of loss functions for heavily imbalanced lesion segmentation,” *Sensors*, vol. 24, no. 6, 2024. [Online]. Available: <https://www.mdpi.com/1424-8220/24/6/1981>
- [28] D. Eigen and R. Fergus, “Predicting depth, surface normals and semantic labels with a common multi-scale convolutional architecture,” in *2015 IEEE International Conference on Computer Vision, ICCV 2015, Santiago, Chile, December 7-13, 2015*. IEEE Computer Society, 2015, pp. 2650–2658. [Online]. Available: <https://doi.org/10.1109/ICCV.2015.304>
- [29] C. H. Sudre, W. Li, T. Vercauteren, S. Ourselin, and M. Jorge Cardoso, “Generalised dice overlap as a deep learning loss function for highly unbalanced segmentations,” in *Deep Learning in Medical Image Analysis and Multimodal Learning for Clinical Decision Support*, M. J. Cardoso, T. Arbel, G. Carneiro, T. Syeda-Mahmood, J. M. R. Tavares, M. Moradi, A. Bradley, H. Greenspan, J. P. Papa, A. Madabhushi, J. C. Nascimento, J. S. Cardoso, V. Bebagiannis, and Z. Lu, Eds. Cham: Springer International Publishing, 2017, pp. 240–248.
- [30] V. Mnih, K. Kavukcuoglu, D. Silver, A. A. Rusu, J. Veness, M. G. Bellemare, A. Graves, M. A. Riedmiller, A. Fidjeland, G. Ostrovski, S. Petersen, C. Beattie, A. Sadik, I. Antonoglou, H. King, D. Kumaran, D. Wierstra, S. Legg, and D. Hassabis, “Human-level control through deep reinforcement learning,” *Nat.*, vol. 518, no. 7540, pp. 529–533, 2015. [Online]. Available: <https://doi.org/10.1038/nature14236>
- [31] L. Jiang, H. Huang, and Z. Ding, “Path planning for intelligent robots based on deep q-learning with experience replay and heuristic knowledge,” *IEEE/CAA Journal of Automatica Sinica*, vol. 7, no. 4, pp. 1179–1189, 2019.
- [32] S. Zhang and R. S. Sutton, “A deeper look at experience replay,” *CoRR*, vol. abs/1712.01275, 2017. [Online]. Available: <http://arxiv.org/abs/1712.01275>
- [33] R. S. Sutton and A. G. Barto, *Reinforcement learning - an introduction*, 2nd Edition. MIT Press, 2018. [Online]. Available: <http://www.incompleteideas.net/book/the-book-2nd.html>
- [34] D. P. Kingma and J. Ba, “Adam: A method for stochastic optimization,” in *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015*,

- Conference Track Proceedings*, Y. Bengio and Y. LeCun, Eds., 2015. [Online]. Available: <http://arxiv.org/abs/1412.6980>
- [35] P. Pandey, D. Pandey, and S. Kumar, “Reinforcement learning by comparing immediate reward,” *CoRR*, vol. abs/1009.2566, 2010. [Online]. Available: <http://arxiv.org/abs/1009.2566>
- [36] F. Zhang, D. Li, S. Li, W. Guan, and M. Liu, “A lightweight tire tread image classification network,” in *2022 IEEE International Conference on Visual Communications and Image Processing (VCIP)*, 2022, pp. 1–5.

Appendix A

Dataset Statistics

These figures have been included to add additional precision to the figures that appear elsewhere in this paper, giving insight into exact values where figures may have been simplified for more informative viewing.

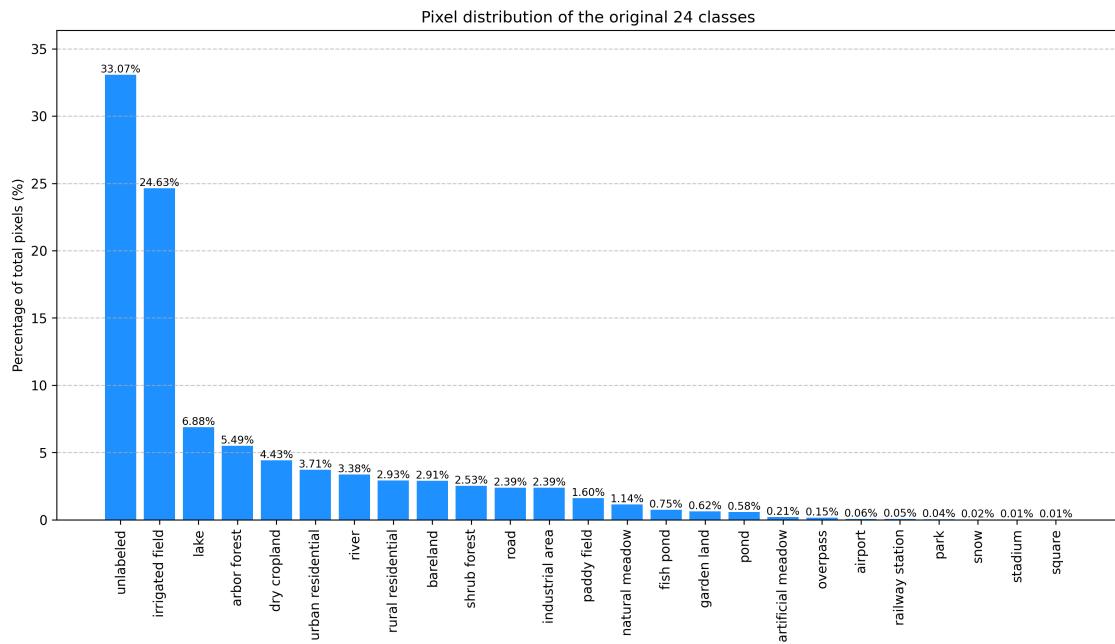


Figure A.1: Shows pixel distribution across the 24 distinct classes in the FBP dataset [4]. It shows how the dataset is mostly dominated by the unlabelled and irrigated field classes which make up over 50% of the dataset.

Appendix B

Additional Metrics

Metric	ResNet50		RestNet34		ResNet18	
	Green	Urban	Green	Urban	Green	Urban
IoU	0.336	0.408	0.792	0.752	0.812	0.808
Precision	0.381	0.469	0.826	0.790	0.871	0.854
Recall	0.493	0.525	0.951	0.939	0.924	0.937
F1-Score	0.453	0.483	0.884	0.858	0.897	0.894

Table B.1: Performance comparison of DeepLabV3+ models with ResNet50, ResNet34, and ResNet18 encoders across segmentation metrics for Green Space and Urban classes. Metrics were computed on the test dataset, revealing that shallower backbones performed better, with IoU scores of 0.812 and 0.808, for the ResNet18 backbone. Whereas a deeper backbone such as ResNet50, had IoU scores of 0.336 and 0.408, showing the problems it had with over-fitting.

B. Additional Metrics

Metric	Green	Urban	UGS
IoU	0.616	0.613	0.051
Precision	0.733	0.657	0.066
Recall	0.794	0.902	0.184
F1-Score	0.762	0.760	0.097

Table B.2: This displays the IoU, Precision, Recall, and F1-Score metrics for the green space, urban area, and UGS evaluations. These metrics show a decreased performance from the binary green and urban area models, with IoU scores of 0.616 and 0.613 respectively, and particularly poor performance from the UGS metric with an IoU score of 0.051

Metric	Green	Urban	UGS
IoU	0.820	0.813	0.372
Precision	0.873	0.857	0.577
Recall	0.931	0.941	0.512
F1-Score	0.901	0.897	0.542

Table B.3: This displays the IoU, Precision, Recall, and F1-Score metrics for the green space, urban area, and UGS evaluations. These metrics show a good performance from the binary green and urban area models, with IoU scores of 0.820 and 0.813, compared to the UGS metric which had an IoU score of 0.372