IDS 705 Project Proposal

UAS Semantic Segmentation for Safe Landing

Drone Vision 2: Attack of the drones

Team Members:
Jabban, Alexander Dany
Lin, Zhanyi
Wu, Yuyou (Pomelo)
Zhu, Yuanjing

Team Number: 02

Introduction

Unmanned aerial vehicles (UAVs), also known as drones or unmanned Aircraft Systems (UAS), are remotely piloted or fully autonomous aerial vehicles equipped with various sensors, cameras, and other instruments to collect data and perform tasks [1]. When initially manufactured in 1916, UAVs were designated to execute military missions and have since been extensively utilized for military surveillance, reconnaissance, and targeted strikes. In the 1990s, the US National Aeronautics and Space Administration (NASA)'s Environmental Research Aircraft and Sensor Technology (ERAST) program marked a significant milestone in the development of protocols and capabilities for using UAS to support scientific research, mapping, and environmental monitoring [2]. At this time, UAV technology became more advanced, and their size, weight, and cost decreased, making them more accessible to non-military organizations. Later with the development of more sophisticated and versatile drones, UAVs' civilian applications have expanded to include fields such as agriculture, journalism, filmmaking, and even package delivery [3]. In 2013, Amazon's chief executive, Jeff Bezos, announced plans to use UAVs for package delivery [4]. Such development led to the creation of Amazon Prime Air, a delivery system that uses UAVs to deliver packages to customers in 30 minutes or less with the benefits of fast and efficient package delivery and reduction of carbon emissions and traffic congestion on roads.

While UAVs or drone delivery offer tremendous benefits and have the potential to revolutionize the logistics field, they face numerous challenges, positing both ethical concerns and technical problems. For civilian and commercial use, drones might violate individuals' privacy, threaten national airspace safety, and cause other problems if malfunction or are used for malicious purposes [5]. The trade-off of drone delivery is a debate between increased delivery efficiency and cost-effectiveness against privacy, safety, and security. Additionally, despite these ethical concerns, drone delivery still encounters some major technical difficulties. For instance, UAVs need to balance the complexity of operating systems and weights. They require sufficient power to operate the propulsion systems, onboard sensors, and communication systems while maintaining lightweight and efficiency to maximize their flight time and payload capacity. One of the largest challenges is the need to correctly identify safe landing places which requires recognizing objects and scenes with short processing time and negligible landing errors. Success landing not only addresses technical problems, but it can also mitigate ethical concerns through differentiating objects, scenes, and humans as well as avoiding airspace crowdedness to maintain safety. To automate object detection and recognition, as well as mapping and surveying of large areas, semantic segmentation is applied for efficient and accurate analysis of high-resolution images and videos captured by UAVs. Semantic segmentation is a computer vision technique that involves the labeling of each pixel in an image to the category of the object to which they belong [6]. Through semantic segmentation, therefore, UAVs can accurately identify and classify

different elements in the environment, such as roads, buildings, and vegetation, which allows UAS to navigate through the environment efficiently and safely [7].

Recent algorithmic development in semantic segmentation has allowed significant improvements in the accuracy of object detection and identification and real-time performance [8]. Depending on the specific landing scenario and the type of drone, i.e., vehicle-based or ship-based drones, the success landing rate can vary from 70% to 97% with different heights and dynamic scenes []. This wide range of accuracy is extremely problematic as it can lead to increased risks of accidents and damage to the drone and surrounding environment. Unreliable landing performance can result in significant delays and potentially life-threatening situations. Therefore, improving landing success rates through advancements in algorithms and best practices is essential. Given the current development, however, ensuring robustness to different environmental conditions and generalization to new environments remains a challenging task. Driven by the potential commercial and environmental benefits of drone delivery, in this study, we will focus on applying deep learning techniques for semantic segmentation to help enable UAS to understand objects in a scene, which is crucial for safe and reliable landing navigations. As enthusiasts and practitioners in the field of computer vision, we are motivated to advance state-of-the-art perception tasks as well as improving the accuracy and efficiency of semantic segmentation involving drone images. Ultimately, we hope to foster the development of fully autonomous UAS and overcome the challenges that come with it.

Goal/Objective

Our primary objective is to enhance the safe landing of UAVs by accurately identifying and labeling objects in a scene, including humans, animals, roofs, concrete, grass, soil, snow, and more. To achieve this, we propose to benchmark semantic segmentation for UAVs using a new dataset of drone images that depict realistic backyard scenarios of varying content, via deep convolutional neural networks. Specifically, our research goals are twofold: firstly, to identify safe landing places for drones, and secondly, to assess the level of difficulty in differentiating various objects and scenes to support the further refinement of object detection and scene identification. By achieving these objectives, we aim to conduct an in-depth analysis to understand the relationship between depths and the accuracy of semantic segmentation, fulfilling our reach goal. Through our study, we aim to contribute to the development of safer and more efficient UAV landing techniques.

Background

Semantic segmentation partitions an image into meaningful and distinct objects, assigning each pixel of the image a label corresponding to the class of the object it belongs to. Unlike traditional image classification tasks that involve identifying the presence of a particular object in an image,

semantic segmentation requires a more granular understanding of the image's content, allowing for a more detailed analysis of the scene. For instance, in autonomous driving (Figure 1), semantic segmentation can assist vehicles in identifying different objects on the road, including pedestrians, cyclists, and other vehicles, enabling them to make better decisions and avoid collisions [9].

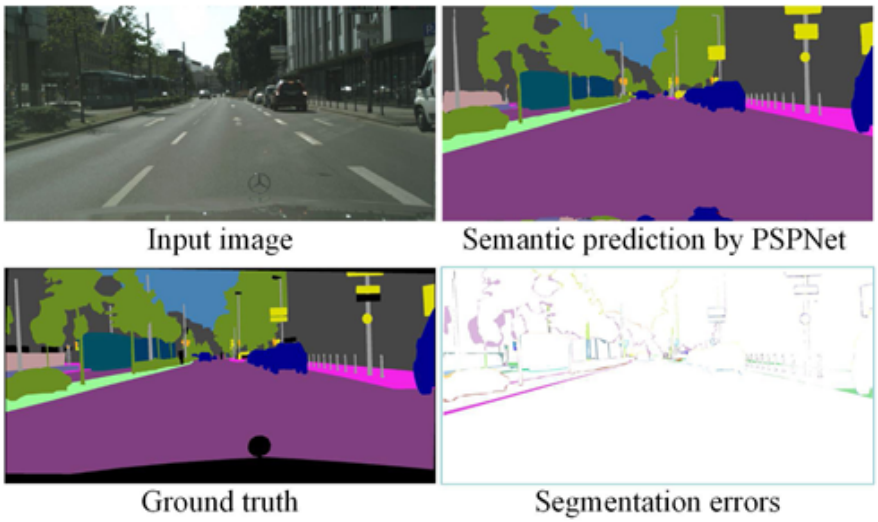

Figure 1 Illustration of semantic segmentation

Several approaches have been proposed to tackle the semantic segmentation problem, ranging from traditional image processing techniques to deep learning-based methods. Early methods involved using hand-crafting features (e.g. edges, texture, and color) and clustering algorithms to segment images. However, these methods often relied heavily on expert knowledge and were limited in capturing complex image structures. In recent years, deep learning-based approaches have achieved remarkable success in semantic segmentation tasks. These methods involve training convolutional neural networks (CNNs) to extract image features and make pixel-level predictions. Building on CNNs, researchers further exploited the architecture advantage of CNNs and created Fully Convolutional Networks (FCNs) [10]. Traditional CNNs utilize a fully connected layer in their final layer, which generates a fixed-size vector indicating the probability of each class for the entire image. In contrast, FCNs replace the fully connected layer with a convolutional layer that produces a spatial map of class predictions for each pixel in the image. This enables FCNs to capture both the local and global context of the image, allowing for more accurate object segmentation.

In 2015, U-net was introduced by researchers at the University of Freiburg and has since become a popular architecture for semantic segmentation analysis [11]. The key feature of the U-Net architecture is the use of skip connections between the encoder and decoder parts of the network, allowing for the preservation of spatial information and the handling of small objects and complex boundaries. In general, UNet improves upon FCNs by incorporating a U-shaped architecture, using skip connections to preserve spatial information, and combining high-level

and low-level features to produce more accurate segmentations. SegNet, introduced in 2016 by researchers at the University of Cambridge, is a more computationally efficient architecture, making it a popular choice for real-time segmentation applications [12]. Specifically, the use of max-pooling indices in the encoder, which is used to upsample the feature maps in the decoder, helps to preserve the original spatial information of the input image while allowing the network to reduce the size of the feature maps. More recent approaches have focused on using attention mechanisms, multi-scale feature fusion, and conditional random fields to expand the receptive field of the network and capture larger contextual information [13-15].

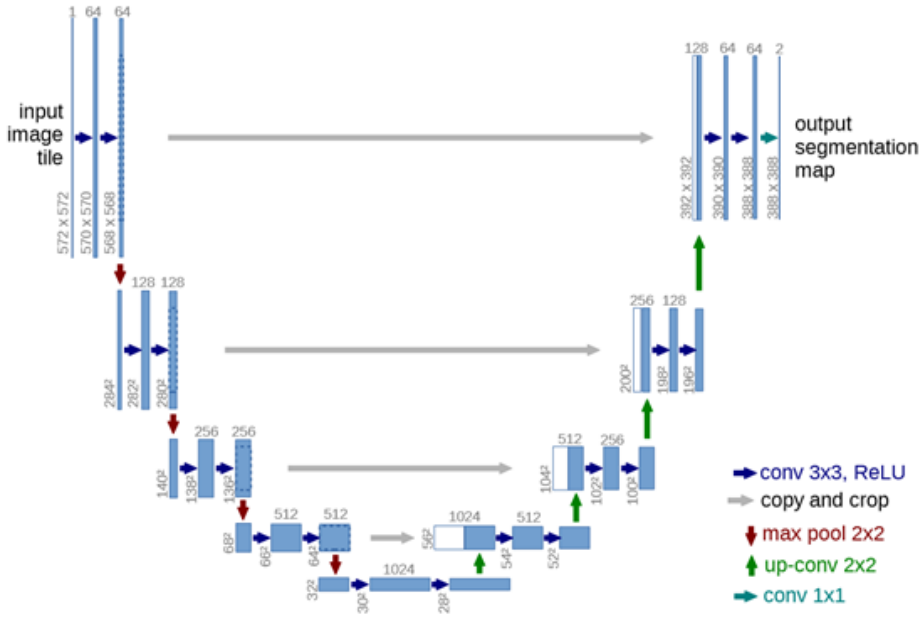

Figure 2. Architecture of U-net for 32×32 pixels in the lowest resolution

We are primarily interested in improving the accuracy of semantic segmentation tasks in our study. Because we are using a new dataset, we intend to apply existing methods to understand the performance of the model in new images. By leveraging existing methods and algorithms, we can identify areas for improvement and explore new approaches to achieve better accuracy in semantic segmentation tasks. Our study will not only contribute to the development of more accurate semantic segmentation models but also advance our understanding of the performance of the models in new and diverse environments. Ultimately, our research aims to pave the way for more accurate and reliable semantic segmentation in a range of applications.

Data

General Description:

Our dataset is provided by AIcrowd which comprises 1787 pairs of images, where each pair consists of an input gray image and its corresponding *uint8* image annotation [16]. The images are stored in two separate folders labeled "*input*" and "*annotation*," respectively. The input images represent the raw data on which the annotation is based, while the annotation images contain labeling for each object or region of interest in the input image at the pixel level.

Both the input and annotation images are stored in *.png* file format, and have the same filename for easy matching. This naming convention also allows for seamless integration with various machine learning pipelines and frameworks, making the dataset highly versatile.

The grayscale input images have a single channel of intensity values that can be used as input data for various image processing and computer vision tasks. The annotations, on the other hand, are stored as *unit8* images, where each pixel corresponds to a particular most likely category for this pixel. This pixel-level labeling can be used to train models to classify different objects or regions of interest in the input images.

Overall, this dataset provides a valuable resource for researchers and practitioners in the fields of computer vision, image processing, and machine learning, enabling the development and evaluation of a wide range of algorithms and models on a diverse set of input images and annotations.

Input file:

The high-resolution ground photo captured by a drone is a crucial component of our drone safety training data. It has a resolution of *2200*1550* pixels, providing a detailed view of the terrain around and below the drone, which allows us to assess the safety of landing areas. Our ultimate goal is to train our drones to automatically analyze the captured image and determine whether the landing area is safe or not.

In Python, the input image is represented as a 2D array with dimensions of *2200*1550*. Each element *(i,j)* of the array corresponds to a particular pixel in the image, with a color code indicating its color or intensity. The pixel at position $e_{(0,0)}$ represents the top-left pixel of the image, while $e_{(0,1)}$ corresponds to the pixel to the right of $e_{(0,0)}$, and $e_{(1,0)}$ represents the pixel immediately below $e_{(0,0)}$. Figure 3 shows an example of the input image.

(a) 

(b)
```
(2200, 1550)
[[29 27 28 ... 40 40 40]
 [28 28 29 ... 39 39 40]
 [30 31 32 ... 40 40 40]
 ...
 [36 42 47 ... 48 50 51]
 [33 40 44 ... 49 51 52]
 [32 38 45 ... 50 51 53]]
```

Figure 3. An example of input image: (a) *.png* format, (b) array format

Annotation File:

The annotation files are an essential component of our training data for the semantic segmentation task. These files, like the input image, are also of size *2200*1550* pixels and serve as labels for the image.

The annotation files contain fully labeled images across 16 distinct categories, with each pixel assigned to its most likely category. There are 17 potential values in each annotation file, with values ranging from 0-15 representing different categories, and the value 255 representing an unknown category.

To illustrate the annotation process, let's consider an example of an annotated section of the image. Suppose a section of the label file has values [[2,2,2,2,2,2], [2,2,3,3,2,2], [2,2,3,3,2,2], [2,2,3,3,2,2], [2,2,2,2,2,2]]. In this case, we can infer that the pixels in this section represent a human on grass. The number 2 represents grass, and the number 3 represents a human. The category of each pixel is represented by the digits in that position.

The annotation files are essential for training machine learning models to perform the semantic segmentation task. By training our models on a labeled dataset, we can enable them to recognize the various categories in the input image and assign each pixel to its corresponding category. This process is crucial for analyzing the safety of landing areas for drones, as it allows us to identify potential hazards such as water bodies, obstacles, and rough terrain. Figure 4 is the example image for the annotation.

(a)  (b)

```
(2200, 1550)
[[255  10  10 ...   1   1   1]
 [255  10  10 ...   1   1   1]
 [255  10  10 ...   1   1   1]
 ...
 [255  10  10 ...  11  11  11]
 [255  10  10 ...  11  11  11]
 [255  10  10 ...  11  11  11]]
```

Figure 4. An example of annotation: (a) *.png* format, (b) array format

Potential Issues and Corresponding Solution:

While our dataset of 1787 observations provides a good starting point, it may be limited in size when training complex models or those with many parameters. To address this issue, we have identified two potential solutions.

Firstly, we can consider cropping the original images into smaller, more manageable sections. This would allow us to effectively increase the size of our training set without compromising on the quality of the data. Given the high resolution of each individual image, we believe this is a reasonable approach to take.

In addition to cropping, we can also utilize data augmentation techniques to further expand our training set. This could involve using variations of the original images, such as rotated versions or those with changes in brightness and contrast. By doing so, we can effectively simulate additional observations, which can help to improve the accuracy and robustness of our models.

Ultimately, we believe that by employing both cropping and data augmentation, we can address the challenge of a limited dataset and achieve more accurate and reliable results.

The dataset for this challenge comprises images of realistic flight footage captured during research and development programs, and not from actual customer deliveries. Additionally, the dataset has been taken measures to ensure that all personal identifiers have been removed from the dataset, thus protecting the privacy of individuals depicted in the footage.

Methods

Our problem has several layers of complexity. We essentially need to perform three fundamental tasks in semantic segmentation, including classification, localization, and object detection. To tackle these problems altogether, we plan on using the **U-Net: Convolutional Network architecture**.

The U-Net architecture can be broken down into two components, an encoder and a decoder. The encoder uses a series of convolutions and pooling to compress the input image creating a decreased resolution representation of the input image. To classify all the pixels in the image, we require an output with dimensions similar to our input. Hence, we will utilize convolutions and upsampling on compressed resolution images with high channel dimensions as the input for the decoder component. In the final convolution, we are using a 1x1 convolution to reduce the channel dimension so we can make predictions on individual pixels. An additional thing to note is that the U-Net architecture is a fully convolutional network thus it has no fully connected layers which greatly reduces the parameter dimensionality for our model [11]. In Ronneberger's original U-net paper, they used 3x3 convolution filters and 2x2 (with stride 2) max pooling filters. We will initially adopt an architecture similar to that of the research paper but adjust some of these in the hyperparameter tuning stage.

Experiments

Planned Experiments Procedures:

1. Baseline model performance tests:
    a. Train the Unet, SegNet, and VIT models on a simple toy dataset with verifiable results.
    b. Evaluate the model performances on the dataset to ensure the neural networks are working as expected and are initialized correctly.
2. Benchmark semantic segmentation and model selection:
    a. Apply the Unet, SegNet, and VIT models on a subset of dataset on the same height and choose the optimal model after hyperparameter tuning.
    b. Utilize the optimal model to evaluate the performance of semantic segmentation, i.e., identifying and categorizing important elements within a scene, including humans, animals, buildings, concrete, grass, soil, snow, and other objects.
    c. Identify safe landing places for drones based on the results from the segmentation process of 17 different categories.
3. Generalize models and compare results:
    a. Apply and adjust the model to perform semantic segmentation on images of different altitudes.

b. Investigate the correlation between depth and segmentation accuracy.

Evaluation Methods for the Outcomes:

1. Evaluate binary accuracy:
    a. Model performance at predicting the correct binary class, i.e. correct label vs incorrect label

2. Evaluate multi-class accuracy:
    a. Model performance on multi-class predictions
    For instance, is asphalt misclassified as water at a higher rate than other classes? This should help us discern which classes are confused the most.
    b. Significance for problem at hand:
    There are certain surfaces/environments that we have extremely high concerns with landing in such as water or busy streets. Thus if certain classes have high misclassification with one of these high risk environments we would want to be informed of that.
3. Evaluate performance at different altitudes.
    a. Changes in semantic segmentation performance at different heights
    Here we will group into three separate height categories, and potentially choose from:
        - Training at one height and testing on another height
        - Training on all the heights vs testing on a single height group
    b. Significance for problem at hand:
    Different drone heights might make it easier or more difficult to distinguish the environment.

Point of Reference for Evaluation:

In the context of the Semantic Segmentation challenge on the AIcrowd platform, mIOU and DICE are evaluation metrics used to assess the performance of models in identifying and labeling objects in drone images.

1. mIOU, ranging from 0 to 1, stands for mean intersection over union, which measures the average similarity between the predicted segmentation mask and the ground truth segmentation mask across all object classes in the dataset. The mIOU score on the leaderboard is around 0.6, with the highest score reaching 0.6940.
2. DICE is another evaluation metric commonly used in image segmentation tasks. It measures the overlap between the predicted segmentation mask and the ground truth segmentation mask, normalized by the total number of pixels in both masks. The DICE score also ranges from 0 to 1, with a score of 1 indicating perfect segmentation. On the leaderboard, the DICE score is around 0.7, and the top score is 0.7807.

Although achieving high metrics of model performance is not the only objective of this project, the performance of other participants' models provide a good reference for us to optimize our model.

Metrics for comparing experiment results

There are two main metrics for this type of predictive task.

1. Pixel accuracy:
    a. This calculates the percentage of pixels that we predicted that match with the ground truth. It is a simple metric and can be useful in many cases
    b. However, one of the drawbacks is that it is not a useful metric for images with imbalanced classes or when one class dominates the image pixels.

2. Average Intersection Over Union:
    a. This is a useful metric for images with imbalance class pixel distributions.
        i. For each class we calculate the intersection of predicted and true class pixels and divide by the union of the two sets.
        ii. After doing this for each class in an image we can average these values
        iii. Even more useful is to apply weighted averages where we weight the accuracy based on risks of precision and recall for a given class.

Roles

Describe the specific roles and responsibilities each team member be taking on for this project.

Task	Lead	Support
Data cleaning and preprocessing	Danny	Yuanjing Zhu
Image cropping and augmentation	Yuanjing Zhu	
Model test on sample dataset		Danny
Model selection and training		Pomelo, Yuanjing Zhu
Application on different altitudes	Pomelo	Yuanjing Zhu
Visualization and analysis	Pomelo	Danny
Write-up, presentation	all	

References

[1] Gupta, S. G., Ghonge, D., & Jawandhiya, P. M. (2013). Review of unmanned aircraft system (UAS). *International Journal of Advanced Research in Computer Engineering & Technology (IJARCET)* Volume, 2.

[2] Long, J., Shelhamer, E., & Darrell, T. (2015). Fully convolutional networks for semantic segmentation. *In Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 3431-3440).

[3] Marcu, A., Costea, D., Licaret, V., & Leordeanu, M. (2019). Towards automatic annotation for semantic segmentation in drone videos. *arXiv preprint arXiv*:1910.10026.

[4] Xiao, X., Zhao, Y., Zhang, F., Luo, B., Yu, L., Chen, B., & Yang, C. (2023). BASeg: Boundary aware semantic segmentation for autonomous driving. *Neural Networks*, 157, 460-470.

[5] Ronneberger, O., Fischer, P., & Brox, T. (2015). U-net: Convolutional networks for biomedical image segmentation. *In Medical Image Computing and Computer-Assisted Intervention–MICCAI 2015: 18th International Conference,* Munich, Germany, October 5-9, 2015, Proceedings, Part III 18 (pp. 234-241). Springer International Publishing.

[6] Badrinarayanan, V., Kendall, A., & Cipolla, R. (2017). Segnet: A deep convolutional encoder-decoder architecture for image segmentation. *IEEE transactions on pattern analysis and machine intelligence*, *39*(12), 2481-2495.

[7] Zheng, S., Jayasumana, S., Romera-Paredes, B., Vineet, V., Su, Z., Du, D., ... & Torr, P. H. (2015). Conditional random fields as recurrent neural networks. In *Proceedings of the IEEE international conference on computer vision* (pp. 1529-1537).

[8] Oktay, O., Schlemper, J., Folgoc, L. L., Lee, M., Heinrich, M., Misawa, K., ... & Rueckert, D. (2018). Attention u-net: Learning where to look for the pancreas. *arXiv preprint arXiv:1804.03999*.

[9] Zhao, H., Shi, J., Qi, X., Wang, X., & Jia, J. (2017). Pyramid scene parsing network. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 2881-2890).

[10] Long, J., Shelhamer, E., & Darrell, T. (2015). Fully convolutional networks for semantic segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 3431-3440).

[11] Ronneberger, O., Fischer, P., & Brox, T. (2015). U-net: Convolutional networks for biomedical image segmentation. *In Medical Image Computing and Computer-Assisted Intervention–MICCAI 2015: 18th International Conference*, Munich, Germany, October 5-9, 2015, Proceedings, Part III 18 (pp. 234-241). Springer International Publishing.

[12] Badrinarayanan, V., Kendall, A., & Cipolla, R. (2017). Segnet: A deep convolutional encoder-decoder architecture for image segmentation. *IEEE transactions on pattern analysis and machine intelligence*, *39*(12), 2481-2495.

[13] Zheng, S., Jayasumana, S., Romera-Paredes, B., Vineet, V., Su, Z., Du, D., ... & Torr, P. H. (2015). Conditional random fields as recurrent neural networks. In *Proceedings of the IEEE international conference on computer vision* (pp. 1529-1537).

[14] Oktay, O., Schlemper, J., Folgoc, L. L., Lee, M., Heinrich, M., Misawa, K., ... & Rueckert, D. (2018). Attention u-net: Learning where to look for the pancreas. *arXiv preprint arXiv:1804.03999*.

[15] Zhao, H., Shi, J., Qi, X., Wang, X., & Jia, J. (2017). Pyramid scene parsing network. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 2881-2890).
[16] AIcrowd (2023), Semantic Segmentation Dataset, Retrieved from: https://www.aicrowd.com/challenges/scene-understanding-for-autonomous-drone-delivery-suadd-23#

Other information

Note: this section is for information purposes and should not be included in your report

How to use this template

1. Click File > Make a copy and delete the instructions replacing text with your report content
2. Follow the formatting instructions contained in this document
3. Share your completed version of the file giving general access to “anyone with the link” and at least “commenter” privileges so that we can add comments and suggestions directly to the text

Length requirements

The proposal does not have any specific length requirements. You are welcome to reuse content written for the proposal for the final report, so if you have a well-written background section, for example, you can reuse that content in the final report saving your team time and effort.

Additional formatting requirements

See the **Final Report Template** for additional formatting requirements