Comparing Deep Learning Approaches to Image Segmentation for Monitoring Deforestation

Jacob Eliason [1] **Rick Holubec** [1] **Chayenne Mosk** [1]

Abstract

Given the ecological significance of the Amazon Rainforest, precise and prompt surveillance is crucial for mitigating the risks associated with biodiversity loss and climate change resulting from deforestation. Deep learning methods can play an important role in this regard due to their ability to produce frequent, accurate estimates. Motivated by the recent advances in image segmentation, this paper assesses multiple state-of-the-art deep learning approaches to accurately identify and track deforestation using 4-band satellite images. Implementing our own approach to image tiling and augmentation, we find that simple models using the FCN and U-Net architectures outperform more complex ones, such as the SegNet, Attention U-Net and DeepLabV3+ models. However, Attention U-Net is shown to be the most efficient model. We do not show any new capacity for image segmentation with satellite imagery using the new Segment Anything model.

1. Introduction

1.1. Problem

The Amazon Rainforest is one of the world's most vital and diverse ecosystems, holding 20% of the planet's liquid freshwater (Assunção & Rocha, 2019) and hosting one in ten species (WWF, 2020). With more than 5.5 square kilometres of dense forests, the biome also plays a fundamental role as a net carbon intaker and climate regulator. Since the early 2000s, local authorities have thus prioritised the protection of the Amazon and substantial investments have been made to combat deforestation (Diniz et al., 2015).

Yet, the Amazon still faces ongoing, large-scale deforestation to this day, mainly due to agricultural and infrastructure expansions, illegal logging, land grabbing and cattle ranching (Vergara et al., 2022). According to PRODES, a deforestation monitoring project launched by the National Institute for Space Research (INPE), an estimated 13,038 square kilometres of Brazilian Amazon was cleared in 2021 – equating to an area of more than eight times the size of Greater London. While this not just jeopardises biodiversity and ecological stability, it could also reach a climate tipping point. Studies anticipate that the Amazon could lose its function as a carbon sink as soon as 2035 (Hubau et al., 2020) and suggest that an additional 5% of deforestation could lead to a non-recoverable transformation into a non-forest ecosystem (Lovejoy & Nobre, 2018). This, in turn, accelerates climate change, making the occurrence of extreme weather events not just more likely but also more pervasive.

1.2. Solutions

Due to the biome's sheer magnitude and complexity, keeping track of its deforestation has traditionally been a challenging task (Diniz et al., 2015). However, recent breakthroughs in satellite imaging technology and deep learning have provided new opportunities to remotely monitor forest loss at a global scale. In particular, novel image segmentation techniques allow for monitoring real-time land use changes and reliable cross-country estimates.

In this regard, Convolutional Neural Networks (CNNs) are excellent at image classification and object identification, but they don't pinpoint object location. To overcome this limitation, Fully Convolutional Networks (FCNs) were introduced by Long et al. (2015), allowing for pixel-level classification, which is known as semantic segmentation. Over time, more complex models based on the FCN architecture have been developed, such as U-Net, which uses additional expansion channels to provide context information at higher resolution levels. SegNet is another adaptation of U-Net that achieves efficient segmentation with fewer parameters. Additionally, Attention U-Net includes attention mechanisms to further enhance accuracy. One of the latest and more advanced FCN models is the Segment Anything Model (SAM), introduced by Facebook in April 2023.

[*]Equal contribution [1]Department of Statistics, London School of Economics and Political Science, London, UK. Correspondence to: Jacob Eliason <j.z.eliason@lse.ac.uk>.

Proceedings of the 39^{th} International Conference on Machine Learning, Baltimore, Maryland, USA, PMLR 162, 2022. Copyright 2022 by the author(s).

1.3. Proposal

This study aims to provide a comparative analysis of five deep learning models – U-Net, SegNet, Attention U-Net, DeepVLab3+ and SAM – for semantic segmentation of deforested areas in the Amazon Rainforest using 4-band satellite data. Specifically, we will evaluate their IoU score, precision, recall and F1-score in identifying deforested areas and compare their computational efficiency.

Moreover, we are among the first to implement SAM in practice. Therefore, this research will also investigate whether more complex state-of-the-art models, such as DeepVLab3+ and SAM, exhibit better performance compared to traditional models, such as U-Net, SegNet, and Attention U-Net.

Furthermore, in contrast to the approach taken by Bragagnolo et al. (2021b) who utilized the original 512x512 pixelled data (obtained from Bragagnolo et al. (2019), we opted to split the data into smaller tiles of size 256x256. This approach allowed us to generate a larger training dataset, which can potentially enhance the model's ability to generalize and improve its performance.

Our study found that the simpler models, FCN and U-Net, outperform the more complex ones, like SegNet and DeepLabV3+, possibly due to overfitting. Among the models tested, Attention U-Net was the most efficient model. Furthermore, splitting the data into smaller tiles may possibly have lead to a worse performance for the more complex models, as it makes it harder for them to capture and maintain the context of the entire image.

2. Related Work

Previous research has demonstrated the efficacy of using deep learning models for semantic deforestation segmentation. John & Zhang (2022) compared an attention-based U-Net with U-Net, Residual U-Net, ResNet50-SegNet and FCN32-VGG16 for detecting deforestation in the Amazon Rainforest and the Atlantic Forest using Sentinel-2 satellite sensor imagery (three-band Amazon, four-band Amazon and Atlantic Forest). The authors showed that the Attention U-Net outperformed the other models, achieving average pixel-wise precision, recall and F1-score of 0.9774, 0.9764 and 0.9769, respectively. They also analysed mask reproductions from each classifier and found that the Attention U-Net was better at detecting non-forest polygons than U-Net. Meanwhile, Andrade et al. (2020) also studied deforestation in the Amazon forest, using Landsat OLI-8 images. They evaluated the effectiveness of DeepLabv3+ based models with other deep learning based methods, namely the Early Fusion and Siamese Convolutional Network. Their findings indicated that the DeepLabv3+ based models provide the best deforestation masks, significantly outperforming the other DL-based methods, in terms of overall accuracy and F1-score. The performance improvements were even more pronounced when the models were trained on limited samples. In the context of detecting Amazon deforestation, the (Attention) U-Net and DeepLabv3+ models were not directly compared against each other in any known study. However, da Costa et al. (2022) evaluated six architectures (U-net, DeepLabv3+, FPN, MANet, PSPNet, LinkNet) with four encoders (ResNet-101, ResNeXt-101, Efficient-net-b3 and Efficient-net-b7) to detect eucalyptus afforestation areas using Sentinel-2 images. They found that – although the differences were not large among the various models – the best performing model was DeepLabv3+ with the Efficient-net-b7 backbone, achieving an IoU of 76.57. However, when Hadinata et al. (2023) adopted the U-Net and DeepLabV3+ for concrete surface damage detection – with complicated surrounding image settings – they found that the U-Net slightly outperforms DeepLabV3+. The models were trained to detect three types of damages, namely cracks, spallings, and voids. U-Net achieved F1 and mIoU scores of 0.7199 and 0.5993, while DeepLabV3+ achieved scores of 0.6478 and 0.5174, respectively. In this paper, we will directly compare the U-Net, Attention U-Net and SegNet with the DeepVLab3+. We will also adopt the FCN, that will serve as a benchmark. Furthermore, we add to the literature by implementing one of the most recent cutting-edge models for image segmentation, namely the Facebook Segment Anything Model (Kirillov et al., 2023).

3. Model Architectures

3.1. U-net

The U-Net architecture was initially developed by Ronneberger et al. (2015) to perform the semantic segmentation of biomedical images and is a specific type FCN (Long et al., 2015). FCNs are different from traditional CNNs because they consist of both a contracting path and an expansive path. The main purpose of the contracting path is to capture contextual information and detect features, while the expansive path is used to enable precise localisation of object boundaries. U-Net has more expansion channels than standard FCNs, which enables the network to convey information context to higher resolution levels. This creates a U-shaped architecture by making the expansion path symmetrical to the contraction path. See Figure 1.

The contracting path in the U-Net architecture is made up of repeated 3x3 convolutional, 2x2 maxpooling and dropout layers, which reduce the spatial resolution of the image while increasing the number of feature maps. The expansive path, on the other hand, consists of multiple transposed 3x3 convolutional and 2x2 upsampling layers that gradually increase the spatial resolution of the image while maintaining the number of feature maps. By doing so, the expansive path helps to restore the original resolution of the image and

refine the localisation of object boundaries. The final layer of the U-Net model consists of a 1x1 convolutional layer that transforms the 64-components resource vector into the required number of classes.

The U-Net architecture has proven to be successful in dense prediction tasks (Wang et al., 2022). One of the primary reasons for its success is that the network has no fully connected layers and solely uses the valid part of each convolution. Furthermore, the U-Net architecture incorporates skip connections that allow the network to pass low-level features from the contracting path to the expansive path. This allows the network to recover fine-grained details lost during the downsampling process in the contracting path.

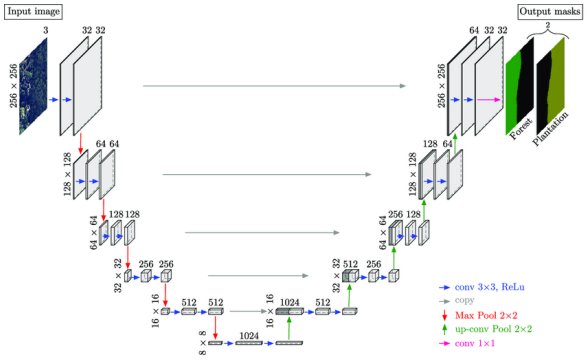

Figure 1. The U-Net architecture (example for 8×8 pixels at the lowest resolution) adapted from Ronneberger et al. (2015). Each box in grey color represents a multi-channel feature map with the number of channels specified at the top of the box. The darker grey boxes denote copied feature maps.

3.2. SegNet

The encoder-decoder architecture of FCNs is also the foundation for SegNet (see Figure 6 in the Appendix), presented by Badrinarayanan et al. (2017). However, SegNet's method for up- and down-sampling of feature maps sets it apart from U-Net. Specifically, SegNet uses a pooling index to save the max-pooling operation's location during the encoder path, allowing for the recreation of the original feature map during the decoder path by upsampling only the maximum activations of corresponding pooling indices. See Figure 7 in the Appendix. This approach reduces the number of trainable parameters needed and makes SegNet faster to train and more memory-efficient compared to U-Net. However, the large number of trainable parameters required for U-Net allows it to better capture fine-grained details (Ronneberger et al., 2015). On the other hand, SegNet tries to retain the fine-gained information in the segmentation differently; while U-Net uses skip connections between the encoder and decoder networks at the *same spatial resolution* level, SegNet uses skip connections directly between the *equivalent layers*.

3.3. Attention U-net

Although the inclusion of skip connections in U-Net and SegNet helps to improve the segmentation performance, it can also result in the extraction of redundant low-level features (Bragagnolo et al., 2021b). These features are often extracted in the initial layers where the feature representation is poor. To overcome this issue, the Attention U-net is created by implementing attention gates at the skip connections (Oktay et al., 2018). These attention gates can selectively suppress the activations in irrelevant regions of the feature maps, thereby reducing the number of redundant features that are extracted and passed on to subsequent layers. An attention gate (see Figure 8 in the Appendix) consists of a squeeze operation, followed by a softmax activation, and a multiply operation. The squeeze operation aims to decrease the number of channels present in the feature maps, which results in the feature maps becoming simpler to handle. The softmax activation function is utilised to evaluate attention weights that identify the significance of each channel in the feature map. Finally, the multiply operation applies these attention weights to the feature maps in order to selectively weigh the most relevant features. The Attention U-Net concatenates each upscaled layer with the output from the attention gates.

3.4. DeepLabV3+

Based on the influential works by Chen et al. (2017a;b; 2018), DeepLabV3+ is a cutting-edge image segmentation technique that has attracted a lot of attention due to its promising image segmentation results on several benchmark datasets (Liu et al., 2021). It extends its predecessor DeepLabV3 by adding a decoder module that particularly improves segmentation performances along object boundaries. Figure 2 provides an overview of the main architecture of DeepLabV3+, which features the following two key concepts:

- *Backbone:* At the base of the DeepLabV3+ architecture is the so-called backbone, a CNN primarily serving as a feature extractor. Even though deep CNNs have shown to be powerful for image segmentation, the repeated application of max-pooling and striding over multiple layers considerably shrinks the spatial resolution. Therefore, DeepLabV3+ uses the building blocks of powerful CNNs such as ResNet50 or Xception as backbone but recovers the spatial resolution using atrous convolution. It is worth mentioning that the application of a modified version of Xception as backbone is unique to DeepLabV3+ and has been proven to perform particularly well for image segmentation. Xception's architecture including the suggested modifications by Chen et al. (2018) is illustrated in Figure 9 in the appendix.

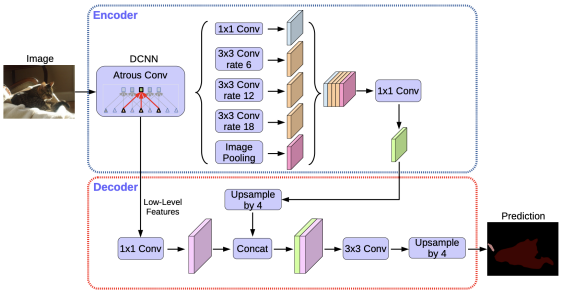

Figure 2. The DeepLabV3+ architecture adapted from Chen et al. (2018). Note that the graph highlights atrous convolution (with $r = 2$ in this case), the ASPP between curly brackets and the encoder-decoder structure.

- *Atrous Convolution:* Atrous or dilated convolution generalises standard convolution by changing the filter's field-of-view. Specifically, by inserting holes (French: 'trous') between filters, the resolution of features can be regulated without increasing the number of parameters. For instance, let y be the output feature map, w be a convolution filter and x an input feature map. Dilated convolution is applied if, for each i, it is given that

$$y[i] = \sum_k x[i + r \cdot k]w[k]$$

  where r denotes the stride such that $r - 1$ intermediate input signals are skipped (for standard convolution $r = 1$). Combining this concept with depthwise separable convolution, multi-scale contextual information is captured better while significantly reducing computational costs.

Put together, the encoder first extracts features from the input image through the backbone which gradually lowers the spatial resolution. The associated resolution reduction ratio between input and output, also known as output stride, is typically set to 16 for semantic segmentation. Subsequently, atrous convolution and Atrous Spatial Pyramid Pooling (ASPP) are applied to extract convolutional features and robustly segment objects at multiple scales. The decoder module then complements the DeepLabV3 architecture with a couple of upsampling and concatenation operations to restore segmentation details.

In this work, DeepLabV3+ is implemented with both the ResNet50 and Xception backbones. However, while the Xception backbone can handle any image, ResNet50 is restricted to a three channel input. Therefore, we consider three model settings: a ResNet50 backbone by reducing the input channels to the conventional rgb format (1), an Xception backbone with the same three input channels as benchmark (2), and an Xception backbone with original four channel input (3).

3.5. Segment Anything

A brand new release in April 2023, the Segment Anything Model (SAM) is a deep learning model designed for image segmentation built upon the concept of promptable segmentation (Kirillov et al., 2023). Unlike other segmentation models, which are designed for specific tasks or a fixed set of tasks, SAM is designed to be an adaptive foundation model that can generalize to various segmentation tasks via prompt engineering.

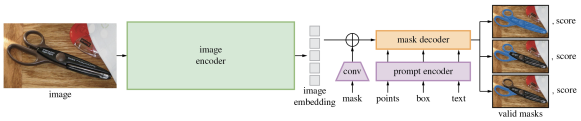

Figure 3. Visual model summary for Segment Anything.

Given its size, SAM has the potential to be suitable for satellite imagery data for image segmentation; its theoretical potential relevance to this project goes without explanation. As SAM enables zero-shot transfer through prompt engineering, it could potentially segment objects and characteristics in satellite images without extensive retraining. However, SAM's performance on satellite imagery would depend in reality on the similarity between its training data and the satellite images under consideration. And in fact, the first limitation listed by its authors is that it "can miss fine structures" and that it struggles to produce boundaries as crisp as task-specific models. The authors suggest on release that biomedical imaging is among the fields likely to continue to outperform SAM despite its strengths. We speculate that, out of the box, the same will be true of satellite imagery.

We seek to answer whether SAM can easily be fine-tuned using a dataset like ours to outperform its standard performance and, potentially, that of customized models like the U-net. As shown in Figure 3, the SAM model consists of three components: an image encoder, a flexible prompt encoder, and a fast mask decoder. The image encoder is large, containing many parameters, and the mask decoder is small, containing few parameters—so to fine-tune it we would likely just want to adjust the weights of the mask decoder.

The source code for SAM gives four input types for its prompt encoder: points, bounding boxes, text, and masks. The first three are most suitable for real-time zero-shot prediction while the latter is the one relevant to our interests. We attempt to run backpropagation on the mask decoder two ways: using our dataset's masks, and then a bounding box containing the entire image, as input prompts. We find neither approach successful; our validation loss resembles something like a random walk through the training process. We point to the complexity of the Segment Anything source

code (and potentially our unfamiliarity with PyTorch, the framework in which it is implemented) as the most likely reasons for our lack of success. However, while we are still very early to working with SAM, there are some early indications on the project's Github page that others have had similar experiences with the model so far. Because it is so early, we decided to still document our unsuccessful efforts as there are not many data points concerning its application to satellite imagery.

4. Training Methods

4.1. Preprocessing

Before training our models, we performed the following preprocessing steps on the 4-band Amazon dataset:

- Image resizing: We attempt to expand the size of our dataset without losing structural information about the underlying patterns by cutting all of the original images into smaller tiles. We used a tile size of 256x256 pixels.
- Data augmentation techniques: To expand our training dataset and improve model generalization, we employed the following data augmentation techniques: 20% of our training images were at random blurred using a Gaussian filter, 50% of our training images were randomly brightened or darkened, and 50% of our training images were randomly made to have higher or lower contrast.
- Shuffling: We shuffled our training dataset to ensure that the model does not learn the order of the training examples.
- Spectral band selection: Some of our model architectures were either incompatible or not easily compatible with the four-band structure represented in our dataset. In those cases, we removed the fourth band.

After performing all necessary preprocessing steps, we export a static version of the training, test, and validation datasets to perform all analysis on to ensure comparability across our choices of architecture.

4.2. Optimizers

For each of our models, we used the Adam optimizer with a learning rate of 0.001. We chose Adam over other optimizers such as SGD and RMSprop because it is computationally efficient, requires little memory, and is well-suited for problems with large datasets and parameters. This choice was also informed by that of the authors of other influential papers performing similar analyses (Kingma & Ba, 2014).

We did not implement momentum or weight decay in our models because we found that they did not improve model performance, and because most of our models performed very well without much fine-tuning.

To better compare all of our models, we use a batch size of 16 and train for 20 epochs. We found that these parameters were sufficient for all of our models to converge; higher and lower batch sizes in particular yielded worse results in terms of both computational efficiency and model performance.

4.3. Overfitting

We incorporated a model checkpoint component into our training process to prevent overfitting. This component saves the model weights after each epoch, and loads the weights from the epoch with the lowest validation loss at the end of training. We do not implement L1 or L2 regularization or dropout in our models. We also do not implement early stopping, learning rate scheduling, or cross-validation.

4.4. Model performance monitoring

We used the binary cross-entropy loss function to measure the difference between the predicted and ground truth masks. This is appropriate for our image segmentation task because we are only interested in two classes: the presence or absence of a "deforested" pixel. We use the following metrics to compare our models:

- Intersection over Union (IoU): The ratio of the intersection and union of the predicted and ground truth masks.
- F1 score: The harmonic mean of precision and recall.
- Precision: The ratio of the number of true positives to the number of true positives and false positives.
- Recall: The ratio of the number of true positives to the number of true positives and false negatives.

Our choices thus follow John & Zhang (2022)'s, who argue that while accuracy is useful as a simple metric, IoU is preferred over accuracy in the context of image segmentation since it evaluates the degree of overlap between the ground truth and prediction rather than just the number of correctly classified pixels. Intersection over Union can be rewritten in terms of true positive (TP), false positive (FP), true negative (TN) and false negative (FN) such that

$$IoU = \frac{TP}{TP + FP + FN}.$$

Precision and recall are two other traditional benchmark measures that are useful as supplementary measures for evaluating the effectiveness of a model. Combining the information contained in each, one may also be interested in the F1-score, which represents their harmonic mean. It follows the form

$$F1 = \frac{2 * Precision * Recall}{Precision + Recall},$$

such that precision and recall contribute equally to the F1-score.

4.5. Implementation

The model was specified using the Keras API with a TensorFlow backend. Training was conducted locally using an Apple M1 GPU. The source code for this work is available online and can be retrieved from our Github repository.

5. Numerical Results

5.1. Goals

The primary goal of our numerical evaluations is to compare various deep learning approaches for image segmentation and assess their performance in deforestation detection using satellite images. As a part of that, this study also investigates whether more complex state-of-the-art models, such as DeepVLab3+ and Segment Anything, exhibit better performance compared to traditional models, such as U-Net, SegNet, and Attention U-Net. To achieve this goal, we have chosen to keep the traditional models relatively simple in terms of layer depth and complexity, while making the latest state-of-the-art models significantly more complex. By doing so, we can accurately assess the efficacy of both simple and complex models in the task of image segmentation.

5.2. Data

We use for our analysis a publicly available dataset containing "4-band" satellite images from the Brazilian Amazon rainforest—primarily in the North between Manaus and Boa Vista, as shown in Figure 4 (Bragagnolo et al., 2021a). The

Figure 4. Location of satellite images (red) shown in context.

dataset consists of 619 images in total, each of which is 512x512 pixels in size. Each image is in `.GeoTiff` format and is associated with a corresponding `.png` binary mask indicating the presence or absence of deforestation. The images contain bands 4, 3, 2 and 8; these correspond to red, green, blue, and panchromatic bands respectively. The panchromatic band is a grayscale image with a higher resolution than the other bands. It's particularly useful for sharpening the other bands, which are lower resolution. The images are converted to byte type and contain values ranging from 0 to 255.

The division of the dataset was performed in advance by the data curators, who assigned 499 images to the training set, 100 images to the validation set, and 20 images to the test set.

5.3. Baseline

In order to benchmark our models, we designate the FCN as our baseline model. The FCN (Long et al., 2015) is a widely used baseline model for semantic segmentation. It was one of the first models to achieve end-to-end pixel-wise semantic segmentation using deep CNNs. To increase segmentation accuracy, the architecture of FCN can be changed to include skip connections, as in the case of U-Net, or to include attention mechanisms, as in the case of Attention U-Net. Despite its success in many segmentation tasks, it has been demonstrated that more sophisticated models, like the ones we are putting forth in this study, outperform the FCN. As a result, in this study, the FCN is considered only for the purposes of comparison.

5.4. Presentation of numerical results

5.4.1. Main results

We present the evaluation metrics for all models in Table 1. Overall, all the models perform reasonably well, achieving scores that are consistently above 0.900. Notably, U-Net stands out as the best-performing model in terms of IoU, Precision, and F1-score. FCN, on the other hand, shows the best performance in terms of recall for both validation and test data. Surprisingly, the relatively simple FCN model consistently achieves very high scores. These results suggest that the simpler models, such as FCN and U-Net, tend to perform better than the more complex ones, potentially due to overfitting. SegNet performs worse in terms of IoU, recall and F1-score compared to the other models. This is likely due to the high number of parameters and the fact that it uses max-pooling, which can result in information loss, especially for images with a high level of detail. Interestingly, this paper's results deviate from the findings by (John & Zhang, 2022), with the Attention U-Net performing worse than the classical U-Net. The underpinnings of this result could be varying data augmentation techniques, in particular the input shape that was reduced from 512x512 to 256x256. Finally, the associated predictions are illustrated graphically in Figure 5.

5.4.2. DeepLabV3+

Surprisingly, all of the DeepLabV3+ model variations returned rather disappointing evaluation metrics and are outperformed by the benchmark FCN. Within the class

Table 1. Quantitative evaluation of five classifiers for classifying forest and non-forest areas within the 4-band Amazon test and validation satellite imagery. The best results are highlighted in bold.

	Validation data				Test data			
Model	IoU	Precision	Recall	F1-Score	IoU	Precision	Recall	F1-Score
FCN	0.9553	0.9708	**0.9836**	0.9772	0.9447	0.9620	**0.9814**	0.9716
U-Net	**0.9617**	**0.9839**	0.9771	**0.9805**	**0.9572**	**0.9815**	0.9747	**0.9781**
SegNet	0.9058	0.9619	0.9395	0.9505	0.9028	0.9518	0.9461	0.9489
Attention U-Net	0.9579	0.9777	0.9793	0.9785	0.9474	0.9682	0.9778	0.9730
DeepLabV3+								
– *ResNet50*	0.9334	0.9827	0.9490	0.9656	0.9184	0.9361	0.9799	0.9575
– *Xception (3 bands)*	0.9135	0.9679	0.9421	0.9548	0.9014	0.9641	0.9328	0.9481
– *Xception (4 bands)*	0.9265	0.9649	0.9588	0.9618	0.9235	0.9618	0.9586	0.9602

of DeepLabV3+ models, the four band Xception backbone achieved the highest out-of-sample scores on the test data, but was outperformed on the validation data by the ResNet50 backbone. Without data augmentation, Xception with three bands performed the best, suggesting that there is not one single best model and questioning the importance of the fourth band (see Table 3 in the appendix).

In an attempt to improve the the model, further sensitivity analyses were performed by testing for alternative feature extractor mechanisms and different output strides but without significant success[1]. A difficulty in this regard is the alignment of the passed image shapes and the exact tracking of where the model fails to capture the details. Along with the small sample size, the characteristics of satellite data and using smaller tiles could be plausible explanations for DeepLabV3+'s failure in this application.

5.4.3. Computational efficiency

Returning back to the overall model evaluations, the computational efficiencies are summarised in Table 2. In particular, the attention U-net turned out to be the most efficient model in terms of training time per epoch whilst having more parameters than the conventional U-Net. In this regard, it is also worth noting that although SegNet has almost 30 times more parameters as U-Net, it requires roughly the same time to train, confirming SegNet's anticipated efficiency.

Table 2. Comparison of the computational efficiency in terms of the number of parameters and training time per epoch.

Model	# of Params	Time/Epoch
FCN	7,774,081	110
U-Net	1,888,833	201
SegNet	29,459,525	203
Attention U-Net	2,583,220	48
DeepLabV3+		
– *ResNet50*	11,852,353	150
– *Xception (3 bands)*	38,207,129	302
– *Xception (4 bands)*	38,207,417	303

6. Conclusion

6.1. Summary

In conclusion, we present a comparative study of five deep learning models − FCN, U-net, SegNet, Attention U-Net, DeepLabV3+ − for the task of deforestation segmentation in the Amazon rainforest using 4-band satellite images. Our results indicate that the simpler models, namely FCN and U-Net, outperform the more complex ones, like SegNet and DeepLabV3+. This is probably due to overfitting and the fact that smaller tiles of data make it harder for the models to capture the context of the entire image. Furthermore, Attention U-Net seems to be the best model in terms of computational efficiency, still giving reasonably high results.

6.2. Limitations

One limitation of this study is the quality of the data labels used. In general, the presence of clouds in satellite images can pose a challenge to accurate labeling, as they may obstruct or distort specific features in the image and affect the reflectance values recorded by satellite sensors. In our specific case, many of the labels inaccurately mark clouds as deforested areas (as is visible in the sample image chosen for Figure 5)—which our models subsequently learn. Used outside this context on imagery from other areas, we would expect our models to continue to incorrectly predict clouded areas were deforested thanks to this erroneous pattern found in the training data.

In addition, accurate inference using our models may be limited to the geographic region of the Amazon rainforest, as

[1]We reduced the number of parameters by a third by choosing other extraction points in the encoder part. Moreover, we tried to implement different stride sizes but that caused issues with the input shape alignment.

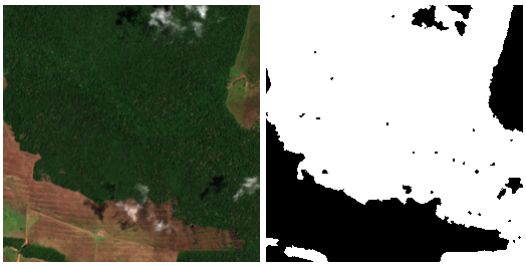

Figure 4. (a) Actual Image (l) and Ground Truth Label (r)

Figure 4. (b) FCN (l) and SegNet (r) Predictions

Figure 4. (c) U-Net (l) and Attention U-Net (r) Predictions

Figure 4. (d) DeepLabV3+ with ResNet50 (l) and with 4-band Xception (r) Predictions

Figure 5. The actual vs. predicted areas subject to deforestation on an out-of-sample satellite image. The biggest differences can be spot on the right-hand-side of the picture. While the FCN and Attention U-Net architectures tend to be overprecise, the DeepLabV3+ with ResNet50 and SegNet lack to capture deforestation overall. For this particular image, the DeepLabV3+ and U-Net return the best predictions.

our training data is sourced from a fairly narrow geographic area. As the identification of deforestation using satellite data is of interest in rainforest climates around the world, a production-quality model should use a training dataset which contains more geographic diversity.

Our study was also limited by computational constraints. If we were to expand our training dataset for any of the reasons mentioned above, we would likely also require additional GPUs for training, as the models trained on our relatively small dataset took in their current state a fairly long time to train. Moreover, analyzing computational efficiency is challenging when conducting research using different computers and computing environments—as our three authors did. While we made every effort to ensure similar conditions across our tests, we acknowledge the possibility that differences in performance or runtime could result from variations in the hardware and software setups used.

6.3. Future Work

This works's findings point to a number of interesting directions for future research. First, it would be intriguing to see how the models would perform on larger datasets. Our training data could have been expanded in terms of greater geographic diversity (sourced from more than one region), temporal diversity (containing observations over a period of time), or spectral diversity (containing more spectral bands than the four represented in our dataset). An alternative to optical satellite imagery, Synthetic Aperture Radar (SAR) also shows promise in monitoring deforestation.

Future work could also explore more creative data augmentation techniques beyond the basic steps (rotation, flipping, brightness) utilized in this study. Recent advancements in data augmentation approaches have shown promising results in improving model performance by generating more diverse and realistic training samples. For example, techniques such as CutMix (randomly masks and replaces patches from different images to create new training samples) and MixUp (interpolates pairs of samples and their corresponding labels) have been shown to be effective in improving model generalization and robustness (Harris et al., 2020).

Finally, we also suggest that further investigation into the Segment Anything Model (SAM) is warranted. As a foundation model designed for generalization across various segmentation tasks, While our initial attempts at fine-tuning SAM for deforestation monitoring using our dataset did not yield satisfactory results, its foundational nature still seems promising for future research. As the model is still in its early stages of development, sharing experiences and collaborating with the broader community working on SAM and its applications could help leverage its potential for deforestation monitoring.

References

Andrade, R., Costa, G., Mota, G., Ortega, M., Feitosa, R., Soto, P., and Heipke, C. Evaluation of semantic segmentation methods for deforestation detection in the amazon. *ISPRS Archives; 43, B3*, 43(B3):1497–1505, 2020.

Assunção, J. and Rocha, R. Getting greener by going black: the effect of blacklisting municipalities on amazon deforestation. *Environment and Development Economics*, 24 (2):115–137, 2019.

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

Bragagnolo, L., da Silva, R., and Grzybowski, J. Amazon rainforest dataset for semantic segmentation, 2019.

Bragagnolo, L., da Silva, R. V., and Grzybowski, J. M. V. Amazon and atlantic forest image datasets for semantic segmentation, 2021a. URL https://zenodo.org/record/4498086.

Bragagnolo, L., da Silva, R. V., and Grzybowski, J. M. V. Amazon forest cover change mapping based on semantic segmentation by u-nets. *Ecological Informatics*, 62: 101279, 2021b.

Chen, L.-C., Papandreou, G., Kokkinos, I., Murphy, K., and Yuille, A. L. Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs. *IEEE transactions on pattern analysis and machine intelligence*, 40(4):834–848, 2017a.

Chen, L.-C., Papandreou, G., Schroff, F., and Adam, H. Rethinking atrous convolution for semantic image segmentation. *arXiv preprint arXiv:1706.05587*, 2017b.

Chen, L.-C., Zhu, Y., Papandreou, G., Schroff, F., and Adam, H. Encoder-decoder with atrous separable convolution for semantic image segmentation. In *Proceedings of the European conference on computer vision (ECCV)*, pp. 801–818, 2018.

da Costa, L. B., de Carvalho, O. L. F., de Albuquerque, A. O., Gomes, R. A. T., Guimarães, R. F., and de Carvalho Júnior, O. A. Deep semantic segmentation for detecting eucalyptus planted forests in the brazilian territory using sentinel-2 imagery. *Geocarto International*, 37(22):6538–6550, 2022.

Diniz, C. G., de Almeida Souza, A. A., Santos, D. C., Dias, M. C., Da Luz, N. C., De Moraes, D. R. V., Maia, J. S., Gomes, A. R., da Silva Narvaes, I., Valeriano, D. M., et al. Deter-b: The new amazon near real-time deforestation detection system. *IEEE Journal of selected topics in applied earth observations and remote sensing*, 8(7): 3619–3628, 2015.

Hadinata, P. N., Simanta, D., Eddy, L., and Nagai, K. Multiclass segmentation of concrete surface damages using u-net and deeplabv3+. *Applied Sciences*, 13(4):2398, 2023.

Harris, E., Marcu, A., Painter, M., Niranjan, M., Prügel-Bennett, A., and Hare, J. Fmix: Enhancing mixed sample data augmentation. *arXiv preprint arXiv:2002.12047*, 2020.

Hubau, W., Lewis, S. L., Phillips, O. L., Affum-Baffoe, K., Beeckman, H., Cuní-Sanchez, A., Daniels, A. K., Ewango, C. E., Fauset, S., Mukinzi, J. M., et al. Asynchronous carbon sink saturation in african and amazonian tropical forests. *Nature*, 579(7797):80–87, 2020.

John, D. and Zhang, C. An attention-based u-net for detecting deforestation within satellite sensor imagery. *International Journal of Applied Earth Observation and Geoinformation*, 107:102685, 2022.

Kingma, D. P. and Ba, J. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

Kirillov, A., Mintun, E., Ravi, N., Mao, H., Rolland, C., Gustafson, L., Xiao, T., Whitehead, S., Berg, A. C., Lo, W.-Y., et al. Segment anything. *arXiv preprint arXiv:2304.02643*, 2023.

Liu, Y., Chu, L., Chen, G., Wu, Z., Chen, Z., Lai, B., and Hao, Y. Paddleseg: A high-efficient development toolkit for image segmentation. *arXiv preprint arXiv:2101.06175*, 2021.

Long, J., Shelhamer, E., and Darrell, T. Fully convolutional networks for semantic segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 3431–3440, 2015.

Lovejoy, T. E. and Nobre, C. Amazon tipping point, 2018.

Oktay, O., Schlemper, J., Folgoc, L. L., Lee, M., Heinrich, M., Misawa, K., Mori, K., McDonagh, S., Hammerla, N. Y., Kainz, B., et al. Attention u-net: Learning where to look for the pancreas. *arXiv preprint arXiv:1804.03999*, 2018.

Ronneberger, O., Fischer, P., and Brox, T. 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, 2015.

Vergara, A., Arias, M., Gachet, B., and Naranjo, L. G. Living amazon report 2022. 2022. URL https://www.worldwildlife.org/publications/living-amazon-report-2022. Accessed on April 17, 2023.

Wang, J., Long, X., Chen, G., Wu, Z., Chen, Z., and Ding, E. U-hrnet: Delving into improving semantic representation of high resolution network for dense prediction. *arXiv preprint arXiv:2210.07140*, 2022.

WWF. Amazon. 2020. URL https://www.worldwildlife.org/places/amazon. Accessed on April 17, 2023.

Statement about individual contributions

All authors contributed equally to the research question formulation and data sourcing. All authors contributed to the determination of the analytical framework and methods. All authors completed independent exploratory data analyses. All authors contributed to the evaluation framework used. All authors contributed to the processing and augmentation script used.

- Mosk: implemented FCN, SegNet and Attention U-Net
- Eliason: implemented tiling function, data saving and loading function, CNN (discarded after preliminary analysis), U-Net without attention, SAM fine-tuning experiment
- Holubec: implemented DeepLabV3+ model with multiple backbones and input shapes, worked on model fine-tuning and optimal model selection (ModelCheckpoint, EarlyStopping)

A. SegNet Architecture

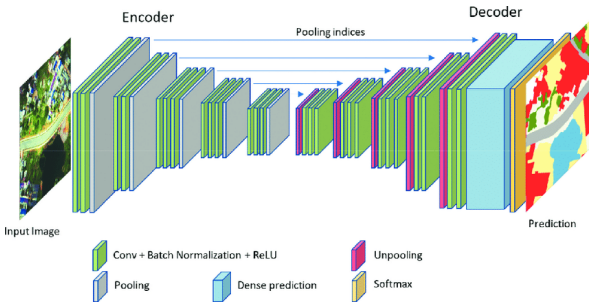

Figure 6. The SegNet architecture introduced by Badrinarayanan et al. (2017). To expand the input information, the decoder relies on the pool indices passed on from the encoder, which helps create a sparse feature map. Subsequently, the decoder applies a convolution operation with a filter bank that can be trained, to fill in the missing details in the feature map.

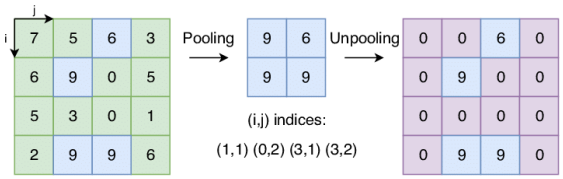

Figure 7. An illustration of how the SegNet decoder works. During upsampling, the max pooling indices at the corresponding encoder layer are recalled for unpooling.

B. Attention Gate

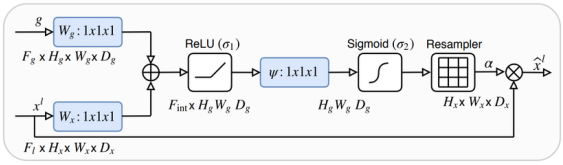

Figure 8. The attention gate illustrated by Oktay et al. (2018) as a diagram, where the input features (x^l) are multiplied by attention coefficients (α) to emphasize important regions. The selection of these regions is determined by analyzing the activations and contextual information provided by the gating signal (g), which is obtained from a coarser scale.

C. Xception Backbone

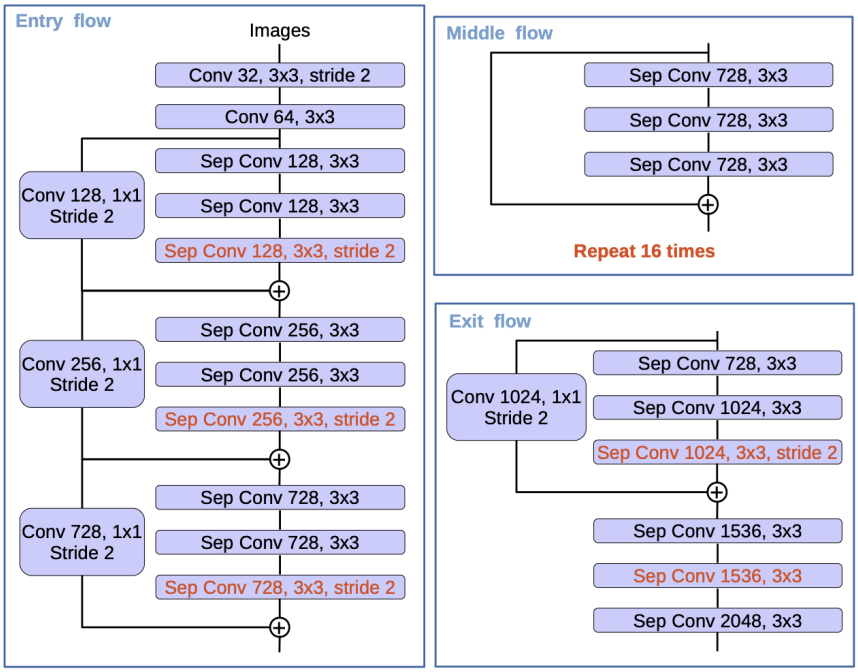

Figure 9. The Xception architecture with adjustments in red according to Chen et al. (2018). The output stride is set to 16 for the Xception backbone.

D. Results without data augmentation

Table 3. Segmentation results without data augmentation on the test dataset.

	Validation data				Test data			
Model	IoU	Precision	Recall	F1-Score	IoU	Precision	Recall	F1-Score
FCN	0.9618	0.9758	**0.9853**	0.9805	**0.9619**	0.9754	**0.9858**	**0.9806**
U-Net	**0.9635**	0.9809	0.9819	**0.9814**	0.9574	0.9777	0.9788	0.9782
SegNet	0.9451	0.9737	0.9699	0.9718	0.9407	0.9701	0.9688	0.9695
Attention U-Net	0.9486	**0.9910**	0.9568	0.9736	0.9453	**0.9890**	0.9553	0.9719
DeepLabV3+								
– *ResNet50*	0.9334	0.9827	0.9490	0.9656	0.9274	0.9768	0.9483	0.9623
– *Xception (3 bands)*	0.9423	0.9728	0.9677	0.9702	0.9386	0.9706	0.9660	0.9683
– *Xception (4 bands)*	0.9353	0.9767	0.9566	0.9665	0.9290	0.9734	0.9532	0.9632