Deep network based approaches to mitigate seasonal effects in SAR images for deforestation monitoring

Carla N. Neves[1], Mabel X. Ortega Adarme[2], Raul Q. Feitosa[2], Juan Doblas Prieto[3], Gilson A. Giraldi[1]

[1] National Laboratory for Scientific Computing, Petrópolis, 25651-076, Brazil - (cneves, gilson)@lncc.br
[2] Pontifical Catholic University of Rio de Janeiro, 22451-900 Rio de Janeiro, Brazil -(mortega, raul)@ele.puc-rio.br
[3] National Institute for Space Research (INPE), São Jose dos Campos, São Paulo 12227-010, Brazil – juan.doblas@inpe.br

Keywords: Deforestation Detection, Synthetic Aperture Radar, Deep Learning.

Abstract

Most deforestation monitoring programs are based on optical images, which are severely affected by clouds, especially in tropical regions. As an alternative, Synthetic Aperture Radar (SAR) data are minimally affected by atmospheric conditions. However, one of the challenges inherent in such approaches is the effect of seasonal rainforest variations over the SAR images. Recently proposed stabilization algorithms mitigate this effect with significant accuracy gains, however, at the cost of higher computational resources, needed to accommodate long temporal SAR image sequences. This work addresses this issue and presents two alternative solutions that attain similar or better accuracy at a much lower computational requirements. One solution is based on a ResUnet, while the second combines an LSTM and a UNet. Both approaches were tested using raw and stabilized pairs as well as raw sequences. Experiments have indicated that both approaches can mitigate the seasonality effect using a much shorter SAR image sequences than modern stabilization methods require. The results with the multitemporal input outperformed those with the preprocessed bitemporal set by 4.3% in Recall, 1.7% in Precision, and 2.7% in F1-Score, also delivering the best deforestation probability maps.

1. Introduction

In Remote Sensing, change detection techniques help to monitor environmental changes by jointly processing two (or more) images of the same geographical area acquired at different dates (Bruzzone and Bovolo, 2012). These processes are essential to monitor the conservation of ecosystems and biodiversity. Change detection using multitemporal remote sensing imagery plays a crucial role in numerous fields of applications, which includes deforestation detection (Ban and Yousif, 2016).

Currently, one of the highest rates of deforestation occurs in South America, with Brazil being the country where the most significant statistics are concentrated. A particular case is the Amazon Forest, as 60% of its area is in Brazilian territory (Amin et al., 2019, Ortega Adarme et al., 2020). Some government projects track deforestation in this region through systematic satellite monitoring. For example, the Amazon Deforestation Monitoring Project (PRODES[1]) provides annual reports about deforestation in the Brazilian Legal Amazon (BLA) since 1988 (Valeriano et al., 2004).

The PRODES procedure and other works that study deforestation detection (Ortega Adarme et al., 2020, Maretto et al., 2020, De Bem et al., 2020) rely on optical data. However, the monitoring of tropical environments with optical imaging is severely limited by cloud cover, particularly during the rainy seasons. An alternative to overcome this observational gap is the use of Synthetic Aperture Radar (SAR) data (Doblas et al., 2020).

The utilization of SAR data for deforestation detection has already been addressed in studies that applied traditional machine learning techniques, such as Support Vector Machine, Random Forest, Maximum Likelihood and distance-based classifiers (Reis et al., 2020, Hethcoat et al., 2020, Diniz et al., 2022, Nicolau et al., 2021). Other studies have addressed that task using SAR data as input for Deep Learning (DL) approaches, like Fully Convolutional Networks (FCN) such as UNet and ResUnet (Ortega Adarme et al., 2020, Ortega et al., 2021, Ortega Adarme et al., 2022), Siamese Networks (Ortega et al., 2021), Multi-Layer Perceptron (Silva et al., 2022), Convolutional Neural Networks (Wahab et al., 2021) and other pretrained DL architectures (Kuck et al., 2021).

SAR measurements, however, are susceptible to signal instability due to atmospheric effects and variations in surface moisture (Ortega et al., 2021, Ortega Adarme et al., 2022). To deal with this problem, Doblas et al. (Doblas et al., 2020) evaluated different SAR data stabilization algorithms for deforestation detection in the Amazon rainforest and concluded that the classification accuracy improves by preprocessing the SAR with methods such as the stabilization proposed in (Reiche et al., 2018).

Another recent study (Ortega Adarme et al., 2022) investigated the potential gains in accuracy brought by stabilization and filtering techniques, such as those used in (Doblas et al., 2020), for detecting deforestation with FCNs. While the filtering operations are intended to reduce the speckle noise in the two images being compared, the stabilization aims to minimize the effect of seasonal variations on SAR images of trees' canopies.

In the study, filtering did not bring significant accuracy gains, being in some cases even deleterious. Stabilization, contrarily, promoted a significant accuracy gain. The authors of (Ortega Adarme et al., 2022) conjectured that the cause of the observed accuracy difference resided in the dataset considered in each case. While the stabilization algorithm uses the information in a time series representing the target area's seasonality, the network tested in (Ortega Adarme et al., 2022) had access only to the stabilized bitemporal image applied to the network's input.

[1] http://www.obt.inpe.br/OBT/assuntos/programas/amazonia/prodes

In this sense, the working hypothesis of this paper is that an FCN can dispense not only with filtering but also with stabilization when determining the deforestation between two dates, provided that it accesses a small sequence of images representing the seasonal variation in the period. The main contributions of this work are:

- two DL-based solutions to minimize the effects of seasonal variations in SAR images in automatic deforestation mapping.
- an experimental analysis of the employed methods using Sentinel-1 data from a sample site of the Amazon forest.
- comparing results of preprocessed bitemporal images and raw multitemporal images in relation to raw bitemporal data to confirm the working hypothesis.

This paper is organized as follows. Section 2 presents an overview of the stabilization algorithm that serves as a baseline in our analysis. Section 3 presents the DL architectures we used in this work. Section 4 reports the experimental setup to validate the proposed architectures. Section 5 shows and discusses the experimental results. The text closes with a summary of the main conclusions drawn from this study.

2. The Stabilization Algorithm

The stabilization method we take as a baseline for comparison with DL alternatives was first presented in (Reiche et al., 2018). It consists of a pixel-wise harmonic fitting approach applied to the SAR backscattering time series.

To determine the stabilized value of an image at a specific location, the algorithm follows three steps: 1) performing a harmonic regression, with no long-term trend component, to fit a sinusoidal function to the backscattering time series at that location, 2) calculating the difference between the original time series and the fitted harmonic function, and 3) adding the median value of the entire time series to the difference obtained in step 2).

Beyond the processing time, the stabilization algorithm is highly memory-demanding because it requires access to a long image time series from the target area (e.g., 88 in the experiments reported in the original paper (Doblas et al., 2020)).

3. Deforestation Detection Approaches

This section describes two Deep Network designs we employed for deforestation mapping from SAR image sequences.

3.1 ResUnet

The first employed architecture builds upon the ResUnet (Ortega et al., 2021), proposed in (Ortega et al., 2021). In short, ResUnet is an FCN that combines residual learning (He et al., 2016) and the UNet architecture (Ronneberger et al., 2015).

The ResUnet input is the tensor $\mathbf{I} \in \mathbb{R}^{H \times W \times C}$ resulting from stacking all images in the sequence $\{\mathbf{I}_i\}_{i=1}^{D}$ along the third dimension, whereby $\mathbf{I}_i \in \mathbb{R}^{H \times W \times \frac{C}{D}}$, H and W denote the spatial dimensions and C being equal to the number of polarizations of each image multiplied by the number D of images in the input sequence.

Our ResUnet-based network has an encoder-decoder architecture (see Fig. 1(a)). It comprises several Residual Blocks (RB - 1(b)) and down-sampling operations that encode the input image into more compact representations.

The decoder comprises a sequence of bilinear upsampling blocks, followed by convolution blocks equipped with a ReLU activation. The decoder output feeds a softmax operator that delivers the posterior class probabilities for each pixel location.

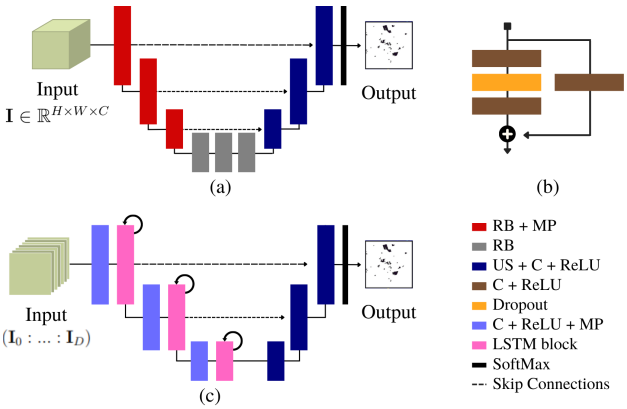


Figure 1. Graphical illustration of the employed architectures. (a) ResUnet architecture. (b) Residual Block (RB). (c) U-LSTM architecture. Symbols: C (Convolution), MP (Max-pooling), RB (Residual Block), US (Up-sampling).

3.2 U-LSTM

The second architecture employed in this work is an adaptation of the network presented in (Papadomanolaki et al., 2019) (see Fig. 1(c)). The proposed deep learning model combines a UNet (Ronneberger et al., 2015) to exploit spatial context from multidate inputs and LSTM blocks to learn patterns along the temporal dimension. The encoding process is repeated for every date independently, so it processes $\mathbf{I}_i$ for $i = 1, \ldots D$ images.

Each convolutional block involves ReLU activation. As for the convolutional operations, we apply 3×3 filters with stride and padding equal to 1. The first convolutional block uses 32 filters and the next two blocks increase the number of filters to twice in each block, also including a 2×2 max pooling operation. All three encoding levels have, in addition to convolution and max-pooling operations, recurring blocks to the temporal relationship among the outputs. This is implemented by replacing the standard fully connected LSTM operations with convolutional structures. In other words, the weights of the recurring operations are no longer simple matrices, but convolutional layers that constitute an end-to-end trainable framework.

After that, the decoder receives the last temporal tensor from the contracting path to upsample it back to its original dimensions with three convolutional blocks similar to those used in the encoder, applying 2×2 upsampling operations instead of max-pooling. The feature map produced by each upsampling operation is concatenated with the calculated temporal pattern of the symmetrical block existing in the encoder part. According to (Papadomanolaki et al., 2019), this is a way to produce more sophisticated features and maintain spatial and temporal knowledge because higher-resolution information is combined with lower-resolution information. Finally, at the end of the model, a 1×1 convolution operation is applied to deliver a tensor with the posterior class probabilities for each pixel location, obtaining the final deforestation map.

4. Experimental Setup

4.1 Dataset

This study employed Sentinel-1 data from a site in the Brazilian Legal Amazon, located in the Pará state, that extends over 115×186 Km². The site is characterized by mixed land cover, mainly composed of dense evergreen forest and pastures (see Figure 2).

For the implementation of the stabilization algorithm, a time series with 88 VH and VV Sentinel-1 images captured between 2019-08-09 and 2020-08-03 was used. Each image with $5,766 \times 9,320$ pixels and a resolution of 10m was downloaded from the Google Earth Engine (GEE) platform(Gorelick et al., 2017).

To build the input data for the DL models described in Section 3, we used seven images captured approximately every second month starting in July 2019 (Fig. 3(a)) and ending in August 2020 (Fig. 3(b)).

Figure 2. Geographical location of the study site in the Pará state, Brazil.

The reference map of the deforestation that occurred in this period is available on the INPE website[2] (Fig. 3(c)). It is worth mentioning that this dataset is highly unbalanced, with only 1.06% of the pixels belonging to the deforestation class, 34.04% corresponding to the past-deforestation class, and 64.9 % to the no deforestation class.

[2] Available online: terrabrasilis.dpi.inpe.br

4.2 Data Preparation and Training

The experimental analysis reported in the next section compares the results obtained with the two DL approaches discussed in Section 3. Three input configurations were used: The first was a bitemporal pair with the first and last available images (a raw pair). The second was the same bitemporal set but with a stabilization preprocessing. Finally, the third configuration was a raw sequence, that is, the set of seven available images, without the stabilization procedure. Table 1 presents the architectures' details.

Architetures	ResUnet	U-LSTM
Encoder	MP(RB(3×3, 32)) MP(RB(3×3, 64)) MP(RB(3×3, 128))	MP (C(3×3, 32)) LSTM (3×3, 32) MP (C(3×3, 64)) LSTM (3×3, 64) MP (C(3×3, 128)) LSTM (3×3, 128)
Bottleneck	3× C(3×3, 128) Dropout (0.5) C(3×3, 128)	-
Decoder	US(C(3×3, 128)) US(C(3×3, 64)) US(C(3×3, 32))	US(C(3×3, 128)) US(C(3×3, 64)) US(C(3×3, 32))
Output	Softmax (C(1×1, #Classes))	SoftMax (C(1×1, #Classes))

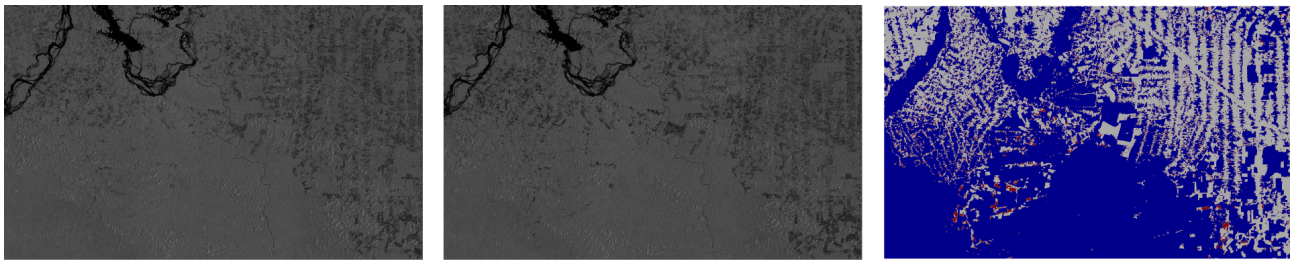
Table 1. Networks Architectures. Symbols: C (Convolution), MP (Max-pooling), RB (Residual Block), US (Up-sampling). The parametrization is (Kernel Height x Kernel Height, Number of filters)

We split each image into 60 tiles of 961×932 pixels and trained the network on image patches of 128×128 pixels cropped from the input tiles, with 70% overlap. We further adopted a cross-validation strategy during the training with six folds. Each tile was part of the test set only once. Then, the final prediction was a mosaic of all test tiles covering the whole image.

We set up the parameters as follows: batch size equal to 32, Adam optimizer with learning rate equal to $1e^{-3}$, and β equal to 0.9, and, to avoid over-fitting, an early stopping strategy with patience equal to 10.

Considering that the dataset is highly unbalanced, we set the focal loss function (Lin et al., 2017) with γ equal to 2 and α equal to to 0.4 for class no deforestation and 0.6 for class deforestation.

Following the PRODES methodology, we ignored the past deforestation class for training, validation, and testing. We took for training only patches having at least 2% of pixels of the deforestation class. In addition, we applied a data augmentation procedure for training and validation; these operations included rotation (multiples of 90º) and flipping (horizontal, ver-



(a) (b) (c)

Figure 3. The raw SAR images at the initial (a) and final (b) dates and ground truth of the deforestation that occurred in the period (c); Legend - gray: past deforestation (1988-2018); red: deforestation (2019-2020); blue: no deforestation.

This contribution has been peer-reviewed. The double-blind peer-review was conducted on the basis of the full paper.
https://doi.org/10.5194/isprs-annals-X-3-2024-261-2024 | © Author(s) 2024. CC BY 4.0 License.

tical) transformations. The threshold to separate the deforestation and no-deforestation classes was 50%.

4.3 Computational Resources

The experiments were conducted on the following system configuration:

- Processor: 32Core AMD Ryzen™ Threadripper™ PRO 5975WX Processor - 3.60GHz 128MB L3 Cache (280W)
- Memory: 8 x 64GB PC4-25600 3200MHz DDR4 ECC RDIMM (512GB total)
- GPU Accelerators: 3 x NVIDIA® RTX A6000 - 48GB GDDR6 - PCIe 4.0 x16
- Operating System: Ubuntu 22.04.2 LTS

The use of all GPUs available on the machine for training was enabled by the Mirrored Strategy function [3] of the TensorFlow deep learning framework. This data parallelism approach is intended to accelerate the training process by allowing a deep learning model to be replicated across multiple GPUs, where each GPU retains a full copy of the model. During training, each replica processes a portion of the training data, and then gradient updates are synchronized between GPUs to update the global model. Using this function further accelerates training and allows for larger models by leveraging memory from multiple GPUs (Pang et al., 2020).

5. Results

The results of each experiment in terms of Precision, Recall, and F1-Score are summarized in Figure 4.

Figure 4 shows that, with ResUNet, the stabilization procedure applied to the pair of input images brought a gain of 3% in terms of Recall, 2% in F1-Score and almost 1% in Precision in comparison with the raw pair. With U-LSTM, the results with the raw pair were closer to the stabilized counterpart, which obtained gains of 1% in Precision and 0.5% in F1-Score, reaching a similar Recall compared to the raw bitemporal input.

Comparing the same raw pair with the raw sequence results, the latter brought a gain of 6,8% in Recall, 2,6% in Precision and 4,7% in F1-Score. With U-LSTM, the raw sequence obtained gains of 4,3% in Recall, 2,1% in F1-Score, but was 0,3% behind in terms of Precision compared to the raw bitemporal input.

In this way, excluding the Precision with U-LSTM, the raw sequence brought greater gains compared to the raw pair than the stabilized pair. The ResUnet variant running on the raw sequence outperformed the stabilized bitemporal images counterpart consistently in terms of all three metrics: the Recall improved by 3.8%, the Precision by 1.7%, and the F1-Score by 2.7%. With U-LSTM, the raw sequence outperformed the stabilized pair in terms of Recall by 4,3% and in F1-Score by 1,6%, but was 1,3 % behind in terms of Precision.

Such results confirm our working hypothesis, namely, that a proper DL architecture having as input an image sequence representing the seasonal variation along the year could waive the stabilization preprocessing and achieve even higher accuracy scores.

[3] tensorflow.org/api_docs/python/tf/distribute/MirroredStrategy

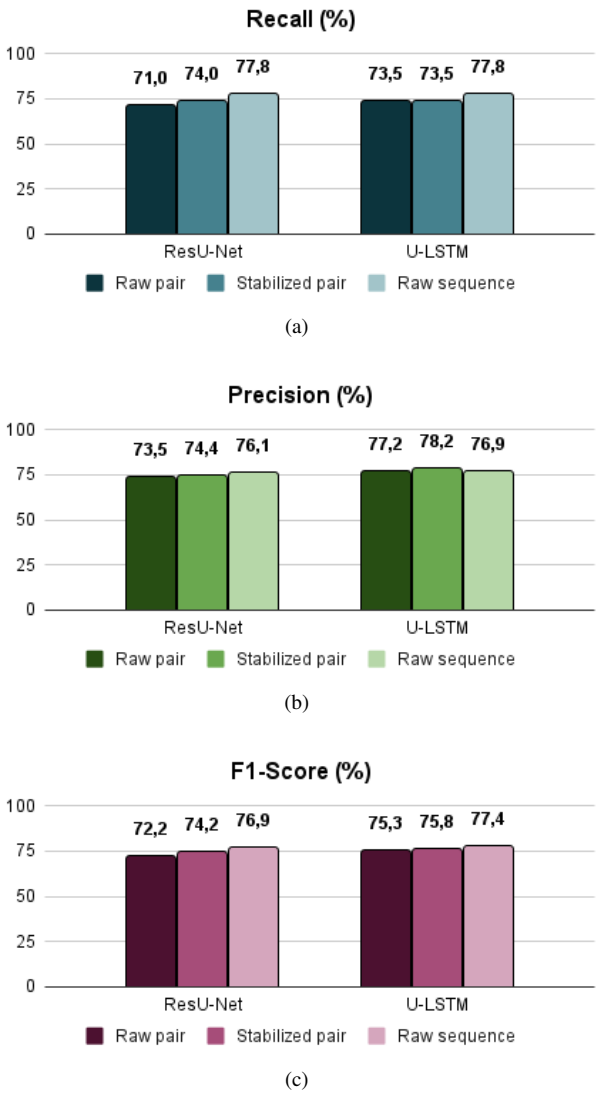


Figure 4. Recall (a), Precision (b) and F1-Score (c) percentages obtained for the target site from ResUnet and U-LSTM using three different input configurations (raw pair, stabilized pair and raw sequence).

Comparing the applied networks, U-LSTM obtained equal or higher results than ResUNet with the same inputs, with exception of the recall with the stabilized pair.

Fig. 5 illustrates the deforestation maps produced in each experiment in snips of the target site. The first row shows the ground truth labels. The following rows show the results obtained with ResUnet and U-LSTM. Each column refers to a different input configuration.

The maps generated with the raw and stabilized image pairs (see Fig. 5(a), (b), (d) and (e)) show more wrongly classified pixels than the maps generated from the multitemporal sequence of raw images (Fig. 5(c) and (f)), corroborating the F1-Score results. In this way, the images in the last column were closer to the reference.

Figure 6 shows the training and inference time for each approach used in the experiments. As expected, the difference

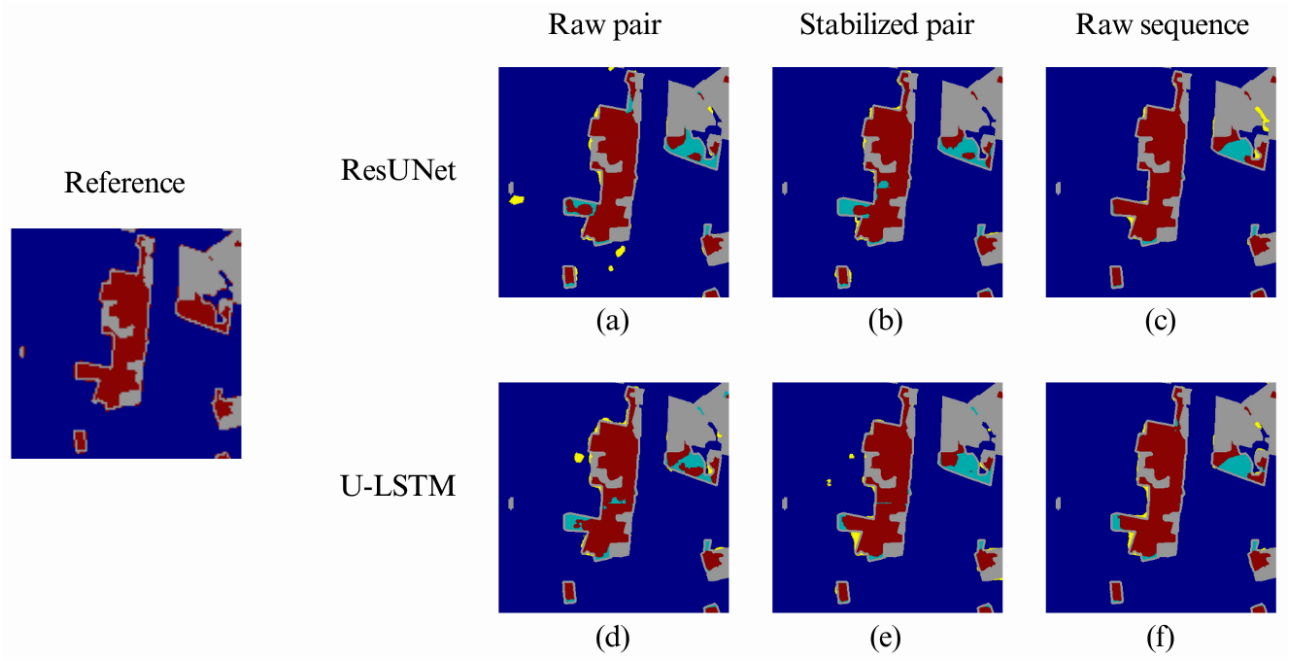
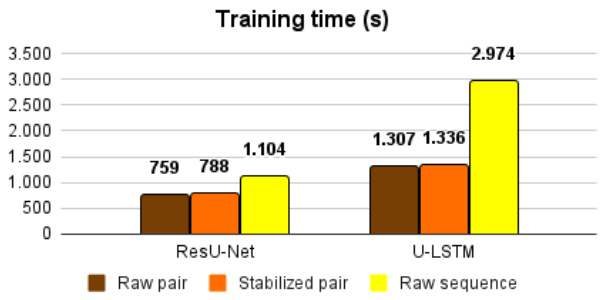
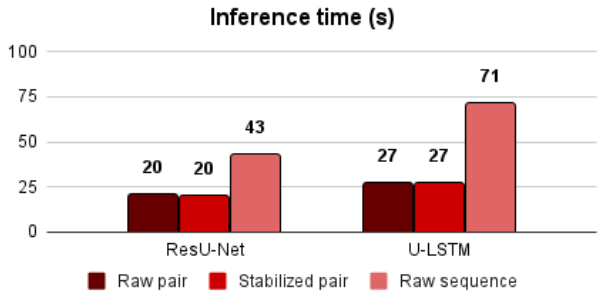


Figure 5. Predicted deforestation maps in a snip from the test set. Legend - past deforestation; deforestation (true positives); no deforestation (true negatives); false positives; false negatives.

in training and inference times between the two configurations that use a pair of dates as input was not expressive. On the other hand, ResUNet working on the temporal sequence doubled the time used for inference and increased the training time in about 40 to 45%, being still in the same order of magnitude as the times measured for both ResUNet configurations.



(a)



(b)

Figure 6. Training (a) and inference (b) time for ResUnet and U-LSTM using three different input configurations.

On the other hand, the complex computation involved in the LSTM blocks inserted between the UNet layers lengthened the training time by more than two times compared to the bitemporal results with U-LSTM and almost three times compared to the ResUNet proposal on the temporal sequence, mainly due to the complex computation. In terms of inference time, the increase was approximately 163% and 65% on the respective comparatives.

However, it is necessary to reemphasize that the stabilization procedure necessitates a significant amount of time and computational resources. As explained in Section 2, the stabilization algorithm requires 88 images for producing the preprocessed pair of images to cope with the seasonal effects over the final model accuracy. Therefore, achieving comparable or superior outcomes without the need for this preprocessing step and using only 7 images instead of 88 can be viewed as a noteworthy development.

6. Conclusion

This study evaluated solutions based on Deep Learning capable of suppressing the effects of seasonality in monitoring deforestation in dense tropical forests from multitemporal SAR images. Specifically, the study presented two solutions, the first based on the ResUnet network and the second combining the UNet network and the LSTM recurrent network, having as input a sequence of raw SAR images acquired at regular intervals in the period of interest.

Experiments conducted on data from the Amazon rainforest in Brazil demonstrated that the architectures can dispense with stabilization algorithms that require long series of temporal images of the target area. The architectures based on ResUnet and UNet/LSTM with multitemporal inputs outperformed the solution based on stabilization algorithms in most accuracy metrics, and ResUNet had less increase in computational load.

As the next step, we consider an ablation study to fine-tune the implemented architectures and assess how the number of

images used to capture the seasonal changes may impact the performance. Another issue worth investigating is that those images may contain relevant information for the semantic segmentation unrelated to the stabilization process. In other words, factors other than those bonded to the compensation of seasonal variations may have contributed to the accuracy gains observed in our experiments. The results encourage tests on other sites to assess the generality of the presented methods.

Acknowledgments

This study was financed in part by the Coordenação de Aperfeiçoamento de Pessoal de Nível Superior – Brasil (CAPES) – Finance Code 001, Conselho Nacional de Desenvolvimento Científico e Tecnológico - CNPq, and Fundação Carlos Chagas Filho de Amparo à Pesquisa do Estado do Rio de Janeiro - FAPERJ.

References

Amin, A., Choumert-Nkolo, J., Combes, J.-L., Motel, P. C., Kéré, E. N., Ongono-Olinga, J.-G., Schwartz, S., 2019. Neighborhood effects in the Brazilian Amazônia: Protected areas and deforestation. *Journal of Environmental Economics and Management*, 93, 272–288.

Ban, Y., Yousif, O., 2016. Change detection techniques: A review. *Multitemporal remote sensing: methods and applications*, 19–43.

Bruzzone, L., Bovolo, F., 2012. A novel framework for the design of change-detection systems for very-high-resolution remote sensing images. *Proceedings of the IEEE*, 101(3), 609–630.

De Bem, P. P., de Carvalho Junior, O. A., Fontes Guimarães, R., Trancoso Gomes, R. A., 2020. Change detection of deforestation in the Brazilian Amazon using landsat data and convolutional neural networks. *Remote Sensing*, 12(6), 901.

Diniz, J. M. F. d. S., Gama, F. F., Adami, M., 2022. Evaluation of polarimetry and interferometry of sentinel-1A SAR data for land use and land cover of the Brazilian Amazon Region. *Geocarto International*, 37(5), 1482–1500.

Doblas, J., Shimabukuro, Y., Sant'Anna, S., Carneiro, A., Aragão, L., Almeida, C., 2020. Optimizing near real-time detection of deforestation on tropical rainforests using sentinel-1 data. *Remote Sensing*, 12(23), 3922.

Gorelick, N., Hancher, M., Dixon, M., Ilyushchenko, S., Thau, D., Moore, R., 2017. Google Earth Engine: Planetary-scale geospatial analysis for everyone. *Remote sensing of Environment*, 202, 18–27.

He, K., Zhang, X., Ren, S., Sun, J., 2016. Deep residual learning for image recognition. *Proceedings of the IEEE conference on computer vision and pattern recognition*, 770–778.

Hethcoat, M. G., Carreiras, J., Edwards, D., Bryant, R. G., Quegan, S., 2020. Detecting tropical selective logging with SAR data requires a time series approach. *bioRxiv*, 2020–03.

Kuck, T. N., Silva Filho, P. F. F., Sano, E. E., Bispo, P. d. C., Shiguemori, E. H., Dalagnol, R., 2021. Change Detection of Selective Logging in the Brazilian Amazon Using X-Band SAR Data and Pre-Trained Convolutional Neural Networks. *Remote Sensing*, 13(23), 4944.

Lin, T.-Y., Goyal, P., Girshick, R., He, K., Dollár, P., 2017. Focal loss for dense object detection. *Proceedings of the IEEE international conference on computer vision*, 2980–2988.

Maretto, R. V., Fonseca, L. M., Jacobs, N., Körting, T. S., Bendini, H. N., Parente, L. L., 2020. Spatio-temporal deep learning approach to map deforestation in amazon rainforest. *IEEE Geoscience and Remote Sensing Letters*, 18(5), 771–775.

Nicolau, A. P., Flores-Anderson, A., Griffin, R., Herndon, K., Meyer, F. J., 2021. Assessing SAR C-band data to effectively distinguish modified land uses in a heavily disturbed Amazon forest. *International Journal of Applied Earth Observation and Geoinformation*, 94, 102214.

Ortega Adarme, M., Doblas Prieto, J., Queiroz Feitosa, R., De Almeida, C. A., 2022. Improving Deforestation Detection on Tropical Rainforests Using Sentinel-1 Data and Convolutional Neural Networks. *Remote Sensing*, 14(14), 3290.

Ortega Adarme, M., Queiroz Feitosa, R., Nigri Happ, P., Aparecido De Almeida, C., Rodrigues Gomes, A., 2020. Evaluation of deep learning techniques for deforestation detection in the Brazilian Amazon and cerrado biomes from remote sensing imagery. *Remote Sensing*, 12(6), 910.

Ortega, M. X., Feitosa, R. Q., Bermudez, J. D., Happ, P. N., De Almeida, C. A., 2021. Comparison of optical and sar data for deforestation mapping in the amazon rainforest with fully convolutional networks. *2021 IEEE International Geoscience and Remote Sensing Symposium IGARSS*, IEEE, 3769–3772.

Pang, B., Nijkamp, E., Wu, Y. N., 2020. Deep learning with tensorflow: A review. *Journal of Educational and Behavioral Statistics*, 45(2), 227–248.

Papadomanolaki, M., Verma, S., Vakalopoulou, M., Gupta, S., Karantzalos, K., 2019. Detecting urban changes with recurrent neural networks from multitemporal sentinel-2 data. *IGARSS 2019-2019 IEEE International Geoscience and Remote Sensing Symposium*, IEEE, 214–217.

Reiche, J., Verhoeven, R., Verbesselt, J., Hamunyela, E., Wielaard, N., Herold, M., 2018. Characterizing tropical forest cover loss using dense Sentinel-1 data and active fire alerts. *Remote Sensing*, 10(5), 777.

Reis, M. S., Dutra, L. V., Sant'Anna, S. J. S., Escada, M. I. S., 2020. Multi-source change detection with PALSAR data in the Southern of Pará state in the Brazilian Amazon. *International Journal of Applied Earth Observation and Geoinformation*, 84, 101945.

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

Silva, C. A., Guerrisi, G., Del Frate, F., Sano, E. E., 2022. Near-real time deforestation detection in the Brazilian Amazon with Sentinel-1 and neural networks. *European Journal of Remote Sensing*, 55(1), 129–149.

Valeriano, D. M., Mello, E. M., Moreira, J. C., Shimabukuro, Y. E., Duarte, V., Souza, I., Santos, J., Barbosa, C. C., Souza, R., 2004. Monitoring tropical forest from space: the PRODES digital project. *International Archives of Photogrammetry Remote Sensing and Spatial Information Sciences*, 35, 272–274.

Wahab, M. A. A., Surin, E. S. M., Nayan, N. M., 2021. An approach to mapping deforestation in permanent forest reserve using the convolutional neural network and sentinel-1 synthetic aperture radar. *2021 Fifth International Conference on Information Retrieval and Knowledge Management (CAMP)*, IEEE, 59–64.