

THE CURIOSITY CUP 2023

A Global SAS[®] Student Competition

Satellite Image Classification for Detecting Deforestation in Romania

T(h)ree Sentinel

ABSTRACT

This paper presents a deep learning approach to monitor deforestation and its legality in Romania. Two datasets consisting of satellite images were used - the first comprised images of the Amazon basin, while the second included images of four regions of interest in the Carpathian Mountains collected by the Sentinel-2 satellite. The images were pre-processed and resized to serve as input for two classification models: a Convolutional Neural Network (CNN) model using SAS Viya and a fine-tuned ResNet50 model in Python. The aim was to predict whether deforestation had occurred or not.

INTRODUCTION

Between 2010 and 2021, *Global Forest Watch* estimated, using satellite images, that Romania lost approximately 204,000 hectares of forest, equivalent to about 286,000 football pitches. Although deforestation is the primary cause of this massive tree loss, the question remains as to how much of this land was legally cut. Declassified documents from the *Romanian National Forest Inventory* revealed that between 2014 and 2018, over 20 million cubic meters of wood were cut down, exceeding the legally permitted amount of 18 million cubic meters.

Deforestation is a global issue, not limited to Romania alone. As a result, numerous authors have attempted to address the problem of detecting illegal deforestation. Kaselimi et al. (2022) approach the issue from the perspective of multilabel classification, training five models - ForestViT, ResNET, VGG, DenseNet, and MobileNet neural networks - to capture land use patterns from satellite images. Of these, ForestViT achieved the highest overall precision (0.80) and recall (0.94). Similarly, Sun et al. (2017) compare the results of three models - classic Deep CNN style (VGG), inception layers (GoogLeNet), and residual layers (ResNet) - with the same objective and methodology. Notably, both studies are based on the same dataset - the Amazon dataset - also used by us for training the models.

PROBLEM

The issue at hand is not limited to deforestation, as it can be carried out sustainably. It is also crucial to consider the legality of the process. The current systems used to monitor deforestation in Romania are susceptible to manipulation as they rely on transport permits acquired through an application with several flaws (the accuracy of the data provided in the notices can only be verified by public authorities). Our proposed approach utilizes models trained on satellite images to monitor forests from above, reducing the likelihood of "intentional errors".

DATA

DATASETS

The initial dataset used to train the models consists of 40,479 images captured from the Amazon basin between January 1, 2016, and February 1, 2017, using Planet's Flock 2 satellites.

The second dataset, utilized as the test set, was gathered from the *Copernicus Sentinel-2 mission* (ESA, 2022), which provides data on alterations in land cover and global forests. We procured satellite images with atmospheric corrections applied to them, covering four distinct regions of interest in the Carpathian Mountains, as depicted in Figure 1. The map tiles were timestamped between May 1, 2022, and July 31, 2022, to capture the forest during its greenest season.

DATA CLEANING AND PRE-PROCESSING

To tailor the Amazon dataset to the specifications of our study and enhance the data quality, we excluded images labeled as habitation, cloudy, and haze. As a result, the sample size decreased to 32,162. We categorized the images into two groups: *deforestation* and *not_deforestation* as can be observed in Figure 2. The deforestation category consists of images from the dataset initially labeled as agriculture, artisanal mine, conventional mine, cultivation, road (forest roads), selective logging, slash burn. The not_deforestation category primarily contains images labeled as primary, along with labels such as water, bare ground, blooming, and blowdown. The original size of the pictures was 256×256 pixels. Using SAS Studio and the action image.processImages from the Image Action Set available in SAS Visual Data Mining and Machine Learning, we resized them to 224×224, as the input layer for the pre-trained models was constructed with that size.

The process for the Sentinel-2 data involved collecting the RGB bands and merging them, followed by normalizing the map tiles to enhance contrast. Each tile was then divided into 4,624 fragments, each with a size of 160x160 pixels, and then resized to 224x224 pixels using Lanczos interpolation (Turkowsky, 1990). The team manually categorized these fragments as either deforestation or not deforestation, and excluded any that did not fit into either category, such as cities or villages.

The final sample sizes of the two datasets, as well as their distribution by label, are presented in Table 1.

Dataset	deforestation	not_deforestation
Train (Amazon data)	9,190 (35.72%)	16,540 (64.28%)
Validation (Amazon data)	2,297 (35.71%)	4,135 (64.29%)
Test (Sentinel data)	1,995 (74.88%)	669 (25.12%)

Table 1. Datasets distribution by label

METHODOLOGICAL FRAMEWORK FOR CNNs

Convolutional Neural Networks are a type of deep learning algorithm that can extract information from an input image to solve various problems, such as classification, segmentation, and object recognition. Typically, a CNN architecture is comprised of multiple convolutional and pooling layers, followed by one or more fully connected layers.

ResNet50 (He et al., 2016) is a type of CNN architecture and a pre-trained model designed for image classification tasks. What is specific to this type of model is that it contains 50 convolutional and fully connected layers, along with other functionalities which improve accuracy. The main components in the ResNet50 architecture are the Residual blocks, which are units of layers that use short layer connections to improve accuracy.

Using SAS Studio through SAS Viya, we built a CNN model and at the same time, using Google Colab and Python we fine-tuned a ResNet50 model to validate the results.

MODEL BUILDING

Before starting to build our models, we performed a few necessary preparation steps. First, a connection to a CAS server was created. Then, the code set up macro variables for the file path of the images, the names of the CAS libraries and tables for the training satellite image data. Using the `image.loadImages` action, we loaded the images. After loading, the images were shuffled to reduce potential bias, and finally, we partitioned the images into two datasets: a training set and a validation set. The partitioning was done using stratified sampling, which sampled 80% of the images for training and 20% for validation while preserving the label frequencies in each set.

For the CNN model itself, we begin by summarizing the training images to determine the offsets by calculating the mean of each of the three-color channels (Red, Green, Blue), which is then used to preprocess the data that feeds into the input layer. We then add several layers using a series of calls to the `deepLearn.addlayer` action:

- Input layer: takes in images of size 224x224.
- Convolutional layer: 10 filters of size 7x7, stride 1, ReLU activation
- Max pooling layer: size 8x8, stride 2
- Fully connected layer: 16 neurons, weight initialization set to Xavier, with bias.
- Output layer: activation softmax, 2 neurons representing the probability of each class.

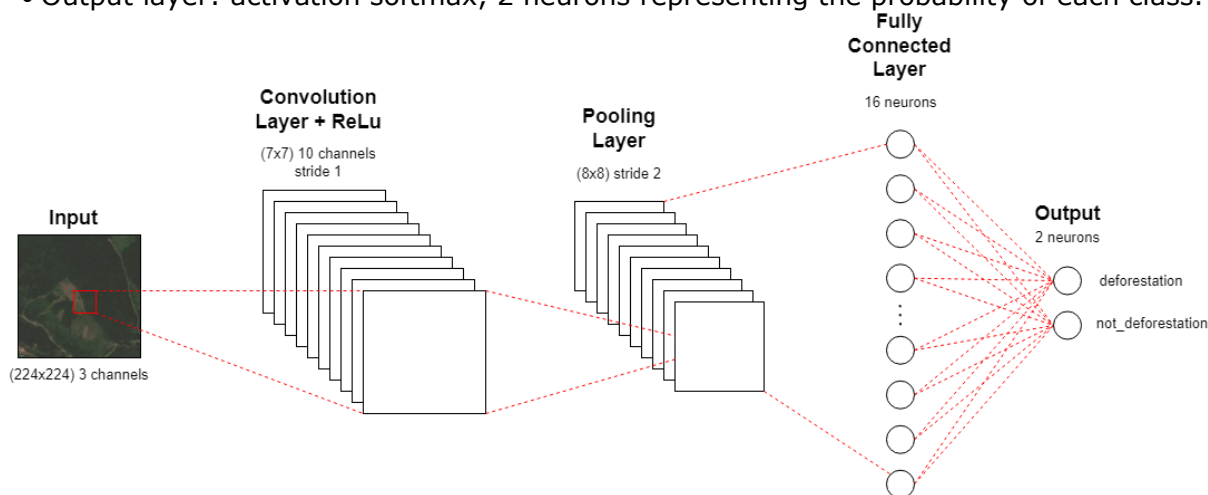


Figure 3. CNN architecture design implemented in SAS Studio

The resulting CNN model with a total number of parameters equal to 2,012,980 was trained using the `deepLearn.dlTrain` action with an ADAM optimizer, a mini-batch size of 2, and a multistep learning rate of 0.0001. The training was set to 5 epochs.

Due to hardware limitations in the academic version of SAS Viya and the lack of GPUs available for model training, this simple CNN architecture was the best solution we could use for a computationally feasible model. Therefore, we decided to fine-tune the ResNet50 model using the same training data and the `fastai` library in Python.

After loading the images, we created batches of 64 images using `DataLoaders` from the `fastai` library. We then partitioned the images using the same stratified sampling approach as before, with 80% of images used for training and 20% for validation. We built a learner using the pre-trained ResNet50 model and found an appropriate learning rate using the `lr_find` function. The learner was fine-tuned on the training dataset, using the base learning rate found earlier,

for a total of 8 epochs. During this time, 4 epochs were "frozen", meaning that only the last fully connected layers were trained.

MODEL VALIDATION

To validate both models a set of metrics was chosen, mainly f1-score and precision score. Precision is the mathematical interpretation of the percentage of correct predictions from all the positive predictions. Recall calculates the ratio of the number of true positives to the total number of positive samples. F1-score is the harmonic mean of precision and recall.

F1-score was selected for its ability to handle class imbalance and provide a balanced evaluation of precision and recall. The Precision-Recall Trade-Off represents the balance between correctly identifying all positive cases and identifying only positive cases.

Based on Table 2, which displays the results on the validation data, the ResNet50 model achieved an f1-score of 0.936 and a precision score of 0.931. In contrast, the CNN model achieved an f1-score of 0.697 and a precision score of 0.693. It is worth noting that the misclassification rate of the CNN model was 0.305, meaning that 30.5% of images were classified incorrectly.

	F1_score	Precision score	Misclassification rate
ResNet50	0.936	0.931	0.082
CNN	0.697	0.693	0.305

Table 2. Model results (weighted average) on validation images from the Amazon dataset.

RESULTS

Driven by the goal of using these proposed models to monitor deforestation in Romania, we needed to evaluate them using accurate data. By using manually labeled satellite images as a test dataset, we were able to determine the effectiveness of our models.

For our ResNet50 model, we applied the test data to the model by first creating a *DataLoader* object and then using the *get_preds* method to obtain an array of class probabilities. The confusion matrix showed that the model made 1774 correct predictions of deforestation and 359 correct predictions of not_deforestation. However, the model also made 310 incorrect predictions of deforestation and 221 incorrect predictions of not_deforestation. The classification report presented in Table 3 provides additional evaluation metrics, indicating an overall weighted average precision of 0.79 and a weighted average f1-score of 0.80.

For the CNN model, the test dataset was applied to the model using the action *deepLearn.dlScore*. The results show that the model made 66 correct predictions of deforestation and 669 correct predictions of not_deforestation, with no incorrect predictions of not_deforestation. However, the model struggled to generalize and made 1929 incorrect predictions of deforestation. The misclassification rate of the model, also shown in Table 3, was calculated as 0.65, indicating that 65% of the predictions made by the model on the Sentinel 2 data were incorrect, compared to the 19.20% for the ResNet50 model that we fine-tuned.

	F1_score	Precision score	Recall score	Misclassification rate
ResNet50	0.80	0.79	0.80	0.1920
CNN	0.0478	0.67	0.0651	0.6522

Table 3. Model results (weighted average) on test images from the Sentinel 2

dataset.

The ResNet50 model achieved significantly better results than the CNN model, indicating that for a complex task such as deforestation image classification, a pre-trained model can be an effective solution. The CNN architecture could potentially be improved by adding more convolutional layers or incorporating techniques such as batch normalization or residual layers. With proper training hardware, the SAS platform can be utilized to create complex CNN architectures for image classification tasks or to fine-tune pre-trained models such as ResNet.

CONCLUSION

The approach we have provided represents a starting point for future forest monitoring systems, as deep learning can play a crucial role in addressing deforestation in Romania. We have shown that with proper training hardware available, it is possible to train models using satellite images from a different part of the world, with the aim of predicting deforestation from a given region. In most cases, the ResNet50 model was able to successfully predict whether an image represents a deforestation event or not, even though the two regions - the Amazonian Forest and the Romanian Carpathians - differ in climate, terrain, and type of forest canopies.

By combining this technology with on-the-ground monitoring, it may be possible to prevent illegal logging and protect Romania's forests and biodiversity more effectively. Authorities can use this in several ways, including identifying deforestation hotspots to focus conservation and rehabilitation efforts, detecting illegal logging, and implementing legal measures, and assessing the impact of different policies.

REFERENCES

Brian Gaines. "How to Develop SAS Code to Train a Deep Learning Model for Image Classification." *SAS Communities Library*. Accessed 10 January 2023. Available at <https://communities.sas.com/t5/SAS-Communities-Library/How-to-Develop-SAS-Code-to-Train-a-Deep-Learning-Model-for-Image/ta-p/663322>.

Christoph Lehermayr, Sebastian Reinhart, and Johannes Kaiser. "Timber mafia and deforestation in Romania." Accessed 05 January 2023. Available at <https://www.balcanicaucaso.org/eng/Areas/Romania/Timber-mafia-and-deforestation-in-Romania-200194>.

European Space Agency. 2022. "Sentinel-2 MSI Level-2A BOA Reflectance [Data set]." In *Sentinel-2 MSI Level-2A BOA Reflectance*. Available at <https://sentinels.copernicus.eu/web/sentinel/sentinel-data-access/sentinel-products/sentinel-2-data-products/collection-1-level-2a>.

Fastai. Accessed 02 February 2023. Available at <https://docs.fast.ai/>

Global Forest Watch. Accessed 02 February 2023. Available at <https://tinyurl.com/mr3y56b2>

He, K., Zhang, X., Ren, S., & Sun, J. 2016. "Deep residual learning for image recognition." In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 770-778).

Joos Korstanje. "The F1 score." *Towards Data Science*. Accessed 15 January 2023. Available at <https://towardsdatascience.com/the-f1-score-bec2bbc38aa6>.

Maria Kaselimi, Athanasios Voulodimos, Ioannis Daskalopoulos, Nikolaos Doulamis, Anastasios Doulamis. 2022. "A Vision Transformer Model for Convolution-Free Multilabel Classification of Satellite Imagery in Deforestation Monitoring". *IEEE Transactions on Neural Networks and Learning Systems*.

Planet. 2017. "Planet: Understanding the Amazon from Space". Dataset from Kaggle. Available at <https://www.kaggle.com/competitions/planet-understanding-the-amazon-from-space/data>

Romania Insider. 2019. "Romania's environment minister confirms massive illegal logging figures". Accessed 02 February 2023. Available at <https://www.romania-insider.com/minister-confirms-illegal-logging-report>

Sun Robert, Castellanos Christian, Nguyen Andrew. 2017. "Deep Multi-Label Classification for High Resolution Satellite Imagery of Rainforests".

Turkowski Ken, Gabriel Steve. 1990. "Filters for Common Resampling Tasks". In *Glassner, Andrew S. (ed.). Graphics Gems I. Academic Press.* (pp. 147-165).

APPENDIX



Figure 1. Sentinel 2 map tiles covering the selected regions from Romania. (ESA, 2022)

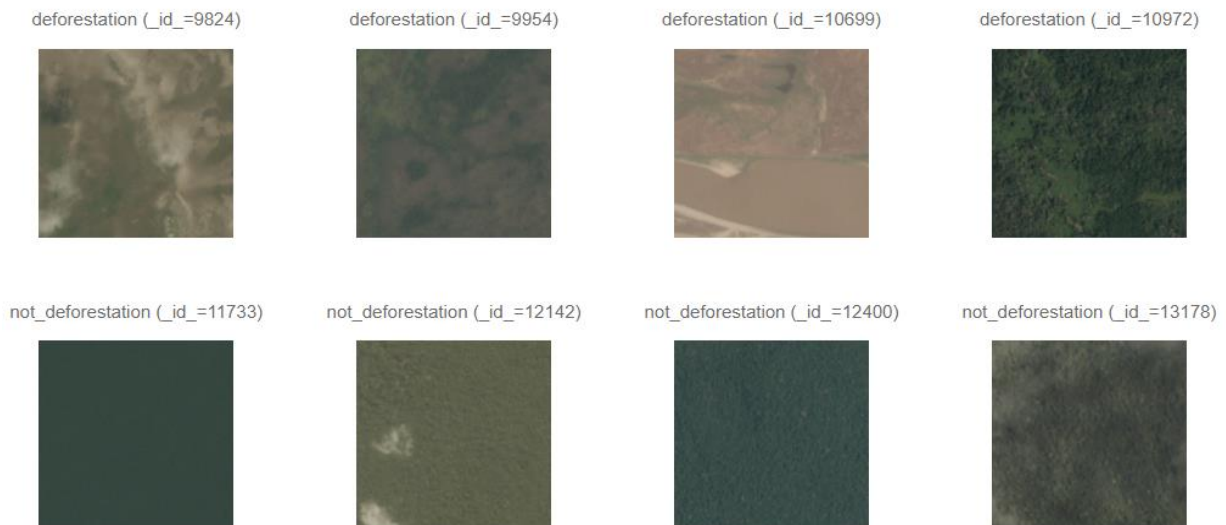


Figure 2. Subset of images loaded in SAS Studio with their respective labels. (Planet: Understanding the Amazon from Space dataset, 2017)