UTRECHT UNIVERSITY

Framework for Deforestation Prediction Using Satellite Images and Neural Networks

by

Lairson Barreto de Melo Filho

A thesis submitted in partial fulfillment for the
degree of Master of Business Informatics

in the
Faculty of Science
Department of Information and Computing Sciences

August 2020

“A arte de viver consiste em tirar o maior bem do maior mal.”

“The art of living consists in getting the greatest good out of the greatest evil.”

Machado de Assis

UTRECHT UNIVERSITY

Abstract

Faculty of Science
Department of Information and Computing Sciences

Master of Business Informatics

by Lairson Barreto de Melo Filho

Human activity is an undeniable factor that increases the total forest area loss. Different NGOs, governments, and private companies are looking for ways to prevent human-driven deforestation. A model that can generate interpretable deforestation predictions is a valuable asset to prevent some causes of forest area loss, such as illegal logging and the creation of pasture or plantation fields. Advances in satellite data analysis and artificial neural networks resulted in different methods of creating such a model. This work lists the most used machine learning techniques to create interpretable predictions. Furthermore, a comparative study of some artificial neural network models with ConvLSTM and Conv3D layers explores how the different techniques lead to different predictions. To compare different models, the validity of the most common evaluation metrics is explored. The unbalanced data distribution leads to the conclusion that precision, recall, F-Score and Balanced Accuracy are the most compelling evaluation metrics for this use case.

Acknowledgements

This document is my final work to obtain a master's degree in Business Informatics from the University of Utrecht. This document is my constant reminder that we can't control everything that happens around us, but we can seize the opportunities that appear if we are prepared to do so. This knowledge was a constant talk of my father (also named Lairson) while growing up. I can see myself as a direct product of my parents, and this makes me smile. Rosalanda, my mother, always said that since I was a kid, she loved me a lot, but she raised me to the world. For this and for prioritizing my education, I thank her. I know that I couldn't be the person I am today, if not my parents, but also my sister. I appreciate not only Lays for being a friend during all my 29 years but also her husband Vagner for giving me tips about academia.

I had the luck to find in Utrecht great friends to share this experience. And found a great path of courses to follow. Unfortunately, I didn't do all courses I wished, but the ones I did were memorable. Technologies for learning, right in the beginning, with my now thesis supervisor Matthieu Brinkhuis, was one of my favorite courses. Other mentions were Requirements engineering with Fabiano Dalpiaz and Adaptive interactive systems with Judith Masthoff.

I want to thank Delloite and WWF teams that I had the pleasure to work with during the last six months. They gave me the support and connections needed during this thesis. On that note, I also thank Matthieu Brinkhuis and Ronald Poppe for the supervising my thesis and giving me ideas to direct my research.

Lastly, I would like to thank Daria Barsukova for her patience. I do not have enough words to explain how lucky I was to have her by my side during the quarantine.

Contents

Chapter 1

Introduction

Since ancient times, for different reasons, human development was always correlated with clearing vast areas of land populated by trees. Those reasons could be, for example, preparing the land for crop production, logging, fiber, building roads, or mining (Abood et al., 2015). The process continues nowadays. Today, the deforestation agents (actors responsible for the process) can be national or international, since the trade of timber and agricultural products are responsible for a substantial amount of tropical deforestation emissions (Pendrill et al., 2019). Most researchers today agree that some degree of deforestation always happened in our society, and there is a direct connection between land cover changes and the natural imbalances that negatively impact our lives today.

There is a strong growing scientific and social movement that tries to analyze and improve the efforts that are preventing the constant destruction of ecosystems (Alencar et al., 2004, Brearley et al., 2019, Mello and Artaxo, 2017, Scholz, 2006). However, since there is a direct connection between deforestation and economic gain (Grieg-Gran, 2006), more than just an ecological approach needs to be taken to plan and prevent deforestation. While solving this arduous task, if state-of-the-art technology is employed, knowledge can be created, awareness can be raised, and local agents can be empowered. To handle predictions based on high amounts of data, machine learning (ML) techniques are the current way to go. Data gathering (and its pre-processing), algorithm selection, model training, and accuracy measures are crucial fragments of the overall machine learning process. There are so many different paths to create a solution that performance comparison between ML algorithms can be a tricky task.

With that in mind, the World Wide Fund for Nature (WWF), Deloitte, and Amazon Web Services joined forces to develop an Early Warning System (EWS) for deforestation. By using the extensive labeled dataset from 2015 to present time, consisting of multispectral images with a monthly update among other data (palm oil mills, road networks, altitude), the main objective of this work is to design and develop a machine learning solution that will predict the future cases new deforestation points in the Indonesian part of the island of Borneo, Kalimantan.

Since human deforestation requires paths to access forest areas, the existence of nearby *roads, highways, and rivers* facilitates the occurrence of deforestation (Barber et al., 2014, Laurance et al., 2001). For example, Barber et al. (2014) directly shows that 94.9% of all deforestation in the Brazilian Amazon occurs within 5.5 km of some type of road or 1.0 km from navigable rivers. Additionally, satellite images can demonstrate where those facilitators are, providing data to understand deforestation.

In 2019, a prototype to predict deforestation was created, showcasing the power of machine learning algorithms in this use case. Despite an overall success with 84% of precision (true positives divided by total positives), some issues were noticed, such as scalability (explained by the amount of data to be stored and processed) and lack of precision (due to a down-sampling after making the predictions).

After studies, interviews, and workshops, WWF concluded that the primary cause of active human deforestation in Indonesia is the palm oil cultivation. Different types of deforestation, such as the growth of pre-existing areas or the creation of new fields, can generate economic value for the locals. Although the data shows a big difference in patterns between types of deforestation, the machine learning model, as it is, has not been able to differentiate them. The main issue being the lack of interpretability, makes it harder to decide what is the best action to take. Possible steps listed by WWF are:

- Supporting law enforcement
- Community engagement
- Company influencing
- Influencing new policy

The sequence of satellite images can be used to predict a future state of the forest. To deal with the complexity of image predictions, a subset of ML is popular among researchers (He et al., 2016, Huang et al., 2017, Krizhevsky et al., 2012). Deep learning (DL) is a specialization of ML which uses artificial neuron networks as building blocks. There are several different DL architectures and techniques, as seen on Chapter 2. These architectures aim to solve domain-specific problems depending not only on the desired outputs but also on the available inputs.

To manage the forest better, it is important that the predictions have interpretability. This allows to trace the deforestation facilitators better, resulting in better-improved prevention policy. Depending on a studied domain, the value of the model's interpretability is different. To understand the reasoning behind a decision might not increase its worth. As for now, most machine learning solutions are seen as a black box, as Sam Harris stated: "you can't really inspect how the algorithm is accomplishing what it is accomplishing." (Harry, 2016). However, deep learning does not always need to be a black box. New techniques for enhancing interpretability, sometimes even with a model-agnostic approach, are being improved every day.

1.1 Problem Statement

The main focus of this thesis project is the development of a framework of different deep learning architectures in order to improve the current EWS solution, resulting in an interpretable temporal prediction. Combining the information found in the last section, we can reach the problem statement by the following line of reasoning:

1. Deforestation is a current problem, and it is hard to stop because of the lack of resources and vast forest areas.
2. Satellite images can provide data about deforestation.
3. There are deep learning methods that can predict changes in satellite data.
4. NGOs could benefit from accurate short-term deforestation predictions.
5. The model needs to have interpretability to perform actions based on the predictions.
6. **Problem statement:**

   There is no guide to aid the creation of a deep learning architecture able to pinpoint and explain the growing deforestation based on satellite images.

1.2 Research Questions

This research combines information about deforestation, satellite images, and deep learning to construct a framework that generates a machine learning architecture that suits this use case. By using the template for design problems (Wieringa, 2014) we can encapsulate the artifact and design problem:

> This research aims to improve deforestation prediction by designing a framework to guide the creation of deep learning architectures such that there is an accurate and interpretable prediction model to facilitate the actions to prevent deforestation.

To limit the scope of this work, the following research questions were defined:

> **MRQ:** How can a framework be developed to aid data scientists in selecting deep learning techniques and process satellite data towards providing accurate and interpretable short-term predictions?

Based on the goal of this research question, four sub-questions were derived to guide the research better.

> **SQ1:** What are the current deep learning architectures and methods suitable for satellite image data prediction?
>
> **SQ2:** What are the design choices to choose and combine techniques to create the deep learning architectures for this use case?
>
> **SQ3:** Which techniques should be used to facilitate the interpretability of the model's results?
>
> **SQ4:** What are the most suitable performance measures for this domain?

1.3 Research Methods

In favor of answering the research questions, this work will execute the design cycle as defined by Wieringa (2014). That means that this research will include a problem investigation, treatment design, and treatment validation. These phases are divided into activities that produce deliverables, as can be seen in figures 1.2 to 1.4 using a process deliverable diagram (PDD) as characterized by van de Weerd and Brinkkemper (2009). The treatment implementation and implementation evaluation, both part of the engineering cycle, will be out of scope for this thesis. Deloitte and WWF will perform them on a future date.

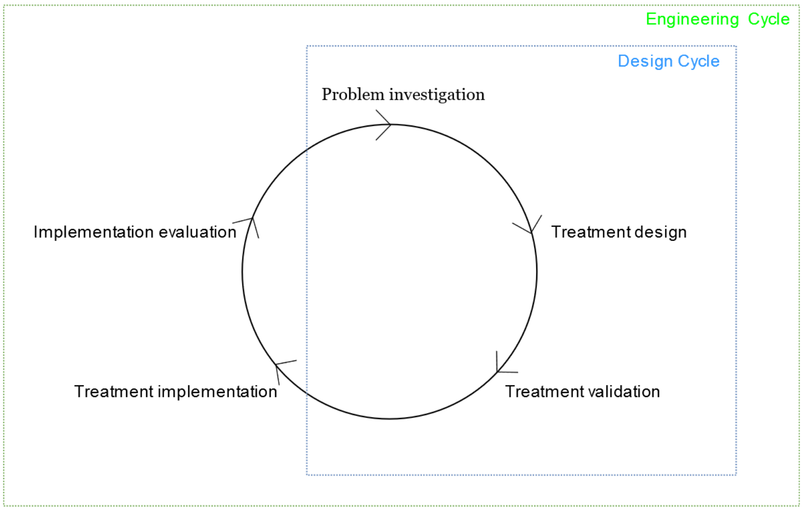


FIGURE 1.1: Research method definition

1.3.1 Problem investigation

The problem investigation phase will start with a literature review, focusing five main topics researched in this order:

1. Deforestation prediction
2. Deep Learning used on satellite data
3. Time-series predictions
4. Interpretable models
5. Performance measurements

The literature review will exclude any topic not related to deep learning techniques, as it is seen as out of scope for this work. The only exception of this is the research about deforestation prediction since a broader view can provide insights on the framework's inputs and outputs.

The reason for a fixed research order is that the review on time series predictions and interpretable models will be directed based on the results of the research on machine learning used on satellite data. This will result in a faster and more directed approach to the topic since interpretable models that cannot be applied to the use case would be logically excluded.

After the literature review, the most prominent deep learning architectures will be listed depending on their availability to work in a cloud-agnostic architecture. This will be the base not only to answer SQ1 but to support the next two phases and, therefore, the answers of the next SQs. Figure 1.2 provides a visualization of the process and deliverables interconnections.

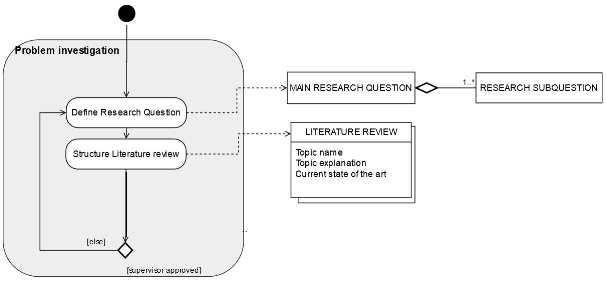


Figure 1.2: PDD displaying the problem investigation

1.3.2 Treatment design

In the treatment design phase, the author will combine suitable deep learning techniques into deep learning architectures to answer SQ2 and SQ3. This result will also provide us with the information required to answer SQ4.

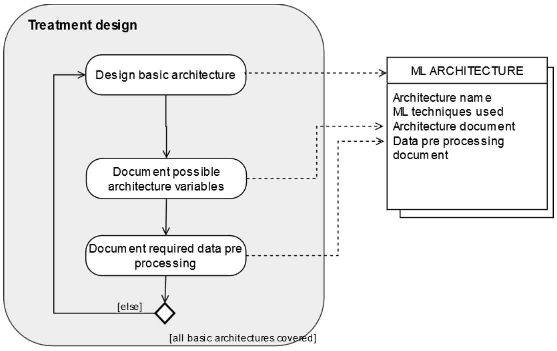

Treatment design
Design basic architecture
Document possible architecture variables
Document required data pre processing
[else]
[all basic architectures covered]
ML ARCHITECTURE
Architecture name
ML techniques used
Architecture document
Data pre processing document

Figure 1.3: PDD displaying the treatment design

1.3.3 Treatment validation

The treatment validation will ensure that the combination of the SQs serves as a base to answer the MRQ. This will be done by using a more significant amount of data and checking if it contributes to WWF goals if implemented. The models generated by the designed architectures will be compared using researched evaluation methods, such as ballanced accuracy, precision, and interpretability.

1.4 Objective, Scope, and Structure

The main objective of this research is to create a methodological framework to generate a DL architecture that contains a model capable of interpretable deforestation predictions.

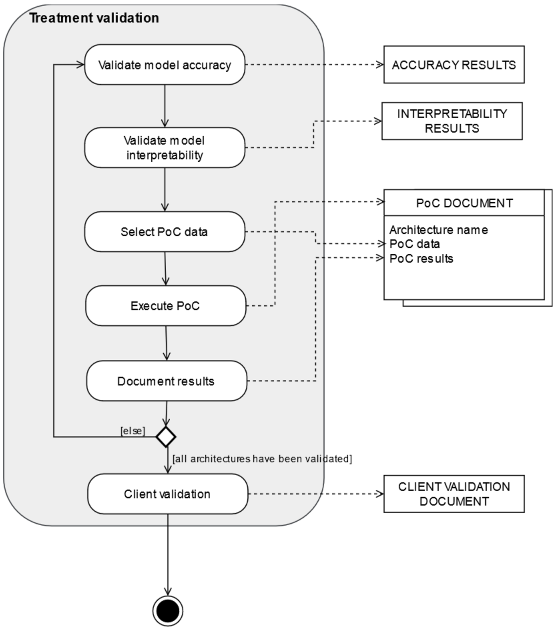


Figure 1.4: PDD displaying the treatment validation

For the scope of this project, data from Kalimantan, the Indonesian part of the island of Borneo, will be used. If successful, this goal of this framework is to be used by WWF and its partners to plan and perform actions to decrease the deforestation rate at different locations, such as Guianas, Brazil, and Malaysia.

When building this solution's architecture, data gathering, pre-processing, and dashboard views are not the focus of this project. Figure 1.5 helps to visualize all the modules that will be presented when building the architecture and which ones will be focused on in this work.

After this introductory section, to be completed with a detailed description of the research approach and research questions, this document will be divided into three main segments:

Part II: Background Theory

In this section, the main ideas that direct this work will be described. Main concepts will define general notions such as deforestation, DL, and interpretability, while The satellite image deforestation prediction task will be an explanation of why it is challenging to predict interpretable changes in satellite data. Next, on Chapter 4 we will have a brief explanation of possible image-based machine learning techniques that can be implemented.

Part III: Practice

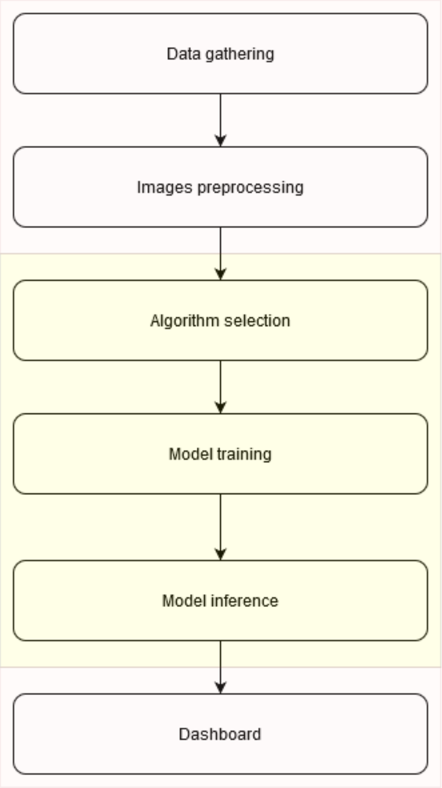


Figure 1.5: Generic architecture. The yellow box limits the scope.

Once all the theoretical background was visited, on this segment, we create and present several viable architectures on chapter 5 and their results on Chapter 6.

Part IV: Conclusion

The response to the research questions, discussions, and future research in chapter 7.

Part II: Background

Chapter 2

Main concepts

This chapter will focus on defining the main concepts that will be used throughout this work. When dealing with some evolving technologies, it is necessary to establish a ground truth.

2.1 Forest

The Kyoto protocol defines a forest area using three parameters. The first one is a minimum land area between 0.05 and 1 hectare. Next, the minimum value of tree cover in this area needs to be between 10 and 30%. Third, the minimum tree height needs to be in the interval of 2 and 5 meters. This means that each country has a window to apply this concept depending on its landscape.

As an unfortunate direct result of this definition, countries can exploit this concept to damage forests enough not to lose the forest status. Sasaki and Putz (2009) warns that the Kyoto protocol definition needs to be revised while explores alternatives. Nevertheless, for this project, the explanation given in the last paragraph will be maintained.

2.1.1 Deforestation

According to Myers (1991), and as the name implies, deforestation is when a forest area is destroyed in a way it can no longer be classified as forest. According to Wibowo and Byron (1999), the same term can be used to address both human activities, addressed at this work's Introduction, and natural causes. Some examples of naturally caused deforestation are, but not limited to, soil erosion, flooding, forest fires, and hurricanes.

While deforestation effects can be perceived with satellite images, the same can not be stated about selective logging.

2.1.2 Degradation

Forest degradation is a process that usually precedes deforestation. The Kyoto protocol does not provide a definition, and there is no universally agreed interpretation (Sasaki and Putz, 2009). This project defines that when a forest loses 10 to 30% of canopy cover on forest area, forest degradation is in course. It is easy to see the effect from the ground, but not so easy using satellites. For this project, radar sensors provide the technology to differentiate both of them, as seen on 2.1. Human-based degradation can also be the result of selective logging, forest usage by guerillas, or drug trafficking (Putz and Redford, 2010). Selective logging is a type of tree removal in which the objective is to retrieve a limited number of marketable tree species (Asner et al., 2005). This modality can damage other trees, affect soil and local fauna. Since it is hard to capture it with satellite images, it will be *out of scope* for this study, unless it can be categorized as forest degradation.

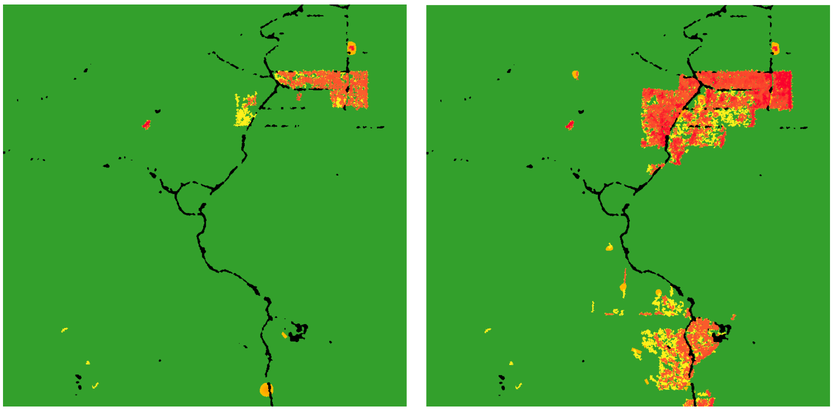


FIGURE 2.1: Radar data examples. Each green pixel is $15m^2$ of forest area. A red pixel corresponds to deforestation and colors between yellow and orange are levels of degradation.

2.2 Deep Learning

Machine Learning (ML) is the area of expertise in Artificial Intelligence that, using data, seeks to learn a given task automatically by experience (Das and Behera, 2017). Mitchell (2006) defines that a machine learns concerning a particular task, performance metric, and type of experience, if the system reliably improves its performance at the task, following the experience.

As stated in the introduction, this thesis will work with a specific type of ML named deep learning (DL). DL consists of the following building blocks.

An artificial neuron is a unit that can receive information from one or several sources as input, transform it, and return an output to any number of targets. DL architectures contain multiple artificial neurons organized in layers, some visible (responsible for input and output), and others hidden (positioned between the visible layers).

As an example, NN can be composed of an input layer of neurons, units that receive the data and calculates a sign to be passed on the next layer of neurons. This procedure repeats itself until an output layer delivers the results (Hassoun et al., 1995). Once the output values are compared to correct answers, error derivatives change the weights inside of the neurons, calculating the next iteration of inputs. This NN can be trained and evolves according to the provided data, training parameters, and type of DL, which can cause changes in its internal architecture. In figure 2.2, we can see a representation of this NN.

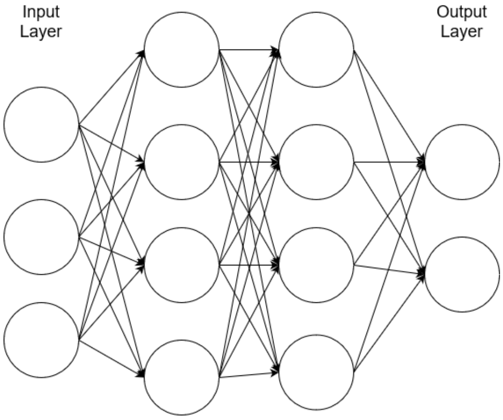


Figure 2.2: Graphic representation of a Neural Network

Advances in NN implementations resulted in the possibility of combining different types of neuron layers with building complex NN architectures. Different connections and behavior mean that some types of layers will be more appropriate for different tasks.

From the current types of NN layers that can be combined to build a DL model, the following will be discussed in this work:

2.2.1 Long Short-Term Memory

Long Short-Term Memory (LSTM) networks are one of the variations built upon the concept of Recurrent Neural Networks (RNN) for this use case. An RNN, explained in Appendix A, is an elegant solution to deal with the time dimension, but the vanishing and exploding gradient problems (Mittelman et al., 2014) badly affect its performance.

To solve this, Hochreiter and Schmidhuber (1997) created LSTM cells to be added at each time step, as depicted in figure 2.3. This internal state will serve as a working memory, which can be updated, deleted, and read when doing the node's calculations. To control the access to the working memory, the node uses three gates with weights and biases. With a similar structure to a neuron in a NN, sigmoid or tanh functions activate the nodes.

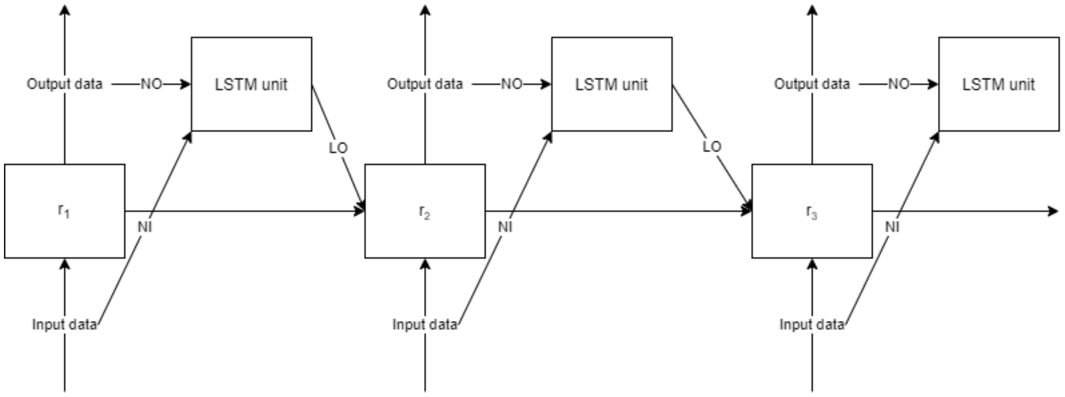


Figure 2.3: Graphic representation of LSTM.

The node fires the write gate *WG* when after analyzing the values of the networks' input data *NI* and the networks' previous output *NO*. This gate will write new data in the working memory. The keep gate *KG* is responsible for maintaining the data. Its input is only the *ND* and calculates how much of the current data should be remembered. The *NI* and the data preserved in the working memory feeds the read gate *RG*, which outputs information that will be directed to the networks' next step as *LO*. Figure 2.4 is a graphical representation of the LSTM unit.

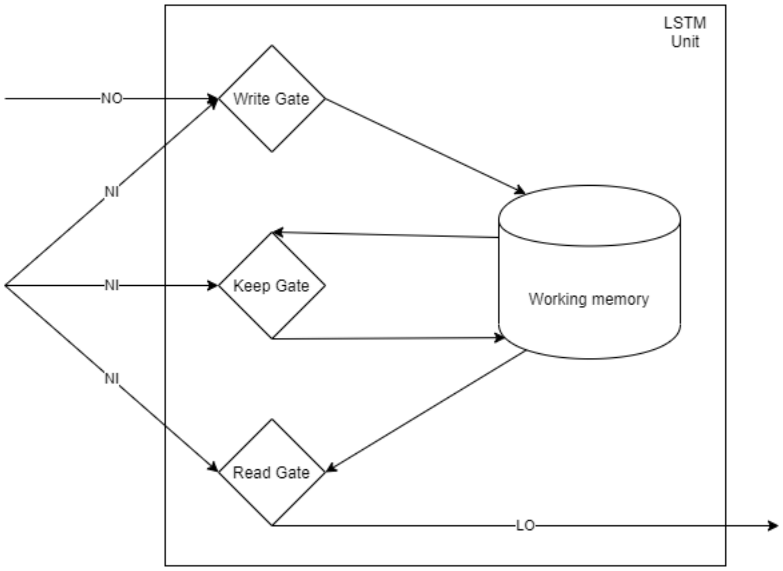


Figure 2.4: Placement of LSTM unit in a RNN.

2.2.2 Convolutional Neural Networks

Convolutional Neural Network (CNN) is a common DL class with a high capacity to process images. Krizhevsky et al. (2012) states that CNNs can "make strong and mostly correct assumptions about the nature of images (namely, stationarity of statistics and locality of pixel dependencies)." They have fewer connections and parameters when comparing to SRTBMs, so they are easier to train.

To capture spatial and temporal dependencies, the CNN usages pattern detection on their convolution layers. The convolution layers, such as any other DL layers, receive input, and produces an output. Each convolution layer contains filters that specialize in pattern detections. The filters (also named kernels) are matrices with defined dimensions depending on the expected input and output. The values inside of the filters are usually initialized with random numbers between 0 and 1. The main objective when training the CNN is to change the filters values to be able to capture the patterns in space and time. The filters *convolve* across the input, generating an output of equal or different sizes, depending on the layer objective. 2.5 exemplifies a filter transforming an input.

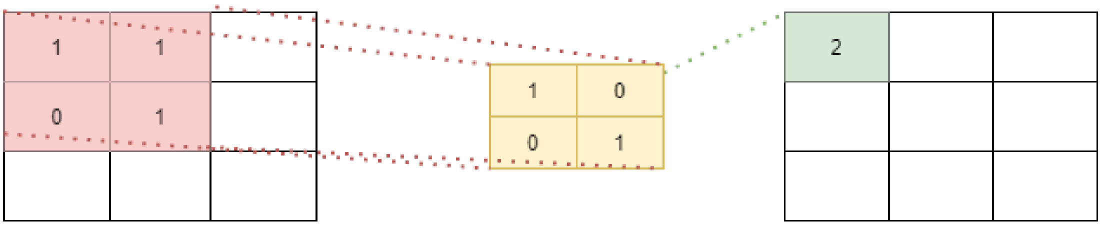


FIGURE 2.5: Exemplification of a convolutional filter. Each cell of the output is the result of the dot product of the filter with a sub-matrix of the same size.

2.2.3 Convolutional LSTM

Alone, an LSTM can understand temporal correlation but does not understand spatial data with it, Xingjian et al. (2015) proposes to have convolutional structures in input-to-state and state-to-state. According to his work, the Convolutional LSTM (ConvLSTM) predicts the next element in a matrix by the inputs and past states of its local neighbors. Since they have fewer neighbors, elements in the border of the matrixes that have less information to predict the future state, therefore, they have a lower prediction potential as explained in subsection 3.2.3.

2.3 Model interpretability

To interpret is to explain the meaning of information, words, or actions. To this day, there is no unique official definition of model interpretability and how to evaluate it (Doshi-Velez and Kim, 2017, Lipton, 2018). To avoid confusion, in this work, the concept will be restrained to the definition given by Ribeiro et al. (2016): "to provide qualitative understanding between the input variables and the response." The interpretability can be part of the ML model or result of a model-agnostic explainer (Ribeiro et al., 2016).

Each model's qualitative understanding will be in the form of correlation. Human-based decision making will be crucial to understand if the correlation leads to causation (Caruana et al., 2015).

Chapter 3

The satellite image deforestation prediction task

This chapter presents an overview of the deforestation satellite image prediction task. Here, based on how a given *area* changes through *time*, it is possible to forecast how the same area will be in a future moment. This chapter contains a list of the most prominent methods to answer the SQ1 and design choices to answer SQ2. It also explores possible evaluation methods to answer SQ4. The choices will further be part of an experiment on chapter 6.

3.1 Forest management

In forest management, time series analysis of satellite images to *understand changes* in land cover was carried out using different techniques, (DeFries and Chan, 2000, Reddy et al., 2013, Rogan et al., 2008, Stone and Lefebvre, 1998, Torahi, 2013), and more recently also to *predict changes* (Ahmadi, 2018, Sales et al., 2017).

As specified at the problem statement, there is no framework to follow when predicting deforestation from satellite images, resulting in a lingering process when comparing and evaluating different techniques of data pre-processing and prediction. Throughout this task's pipeline, there are options to be selected that can impact the final model performance. The first one regards the algorithm selection, and to do this, we need to understand more about the task itself.

3.2 Satellite images

Although the data gathering and pre-processing are out of the scope for this work, it is essential to be clear which data this work will use and some possible limitations for the general scope of the satellite image deforestation prediction task.

3.2.1 Coordinate system

A coordinate system expresses the position of geographic features within a defined *datum* (a mathematical representation of the shape of the earth's surface) and *map projection* (transformations of the spheroidal shape of the earth) (Booth and Mitchell, 2001). To locate the satellite images using a unique geographic dataset, a single coordinate system needs to be defined.

There are two common types of coordinate systems. The *Geographic* system uses degrees to capture the three-dimension surface of the earth. In order to display data into a two-dimensional surface, a *Projected* system is more suitable. It maintains constant angles, lengths, and areas across two dimensions. A trade-off is found on distortions common in large areas of projected coordinate systems, especially if that area is not close to the equator. The available data from WWF uses a projected coordinate system. Since the equator passes through the island of Borneo and the areas of each satellite image are not considered significant within the global scale, the use case images do not have significant image distortions.

3.2.2 Data description

Currently, the available database consists of 6 adjacent areas in the Indonesian part of Borneo. Each area is represented by large TIFF images with around 20 000 square pixels, and about 100-time steps for an individual location. Each pixel represents 15 squared meters. Figure 3.1 is an example of a large TIFF image.

The Image is divided into three main classes, represented by colors. The green pixels represent the forest, red pixels are deforestation, and a spectrum from yellow to orange represents degradation. Lastly, the black pixels are rivers, lakes, roads, sea, and other land uses dated before the first image collection.

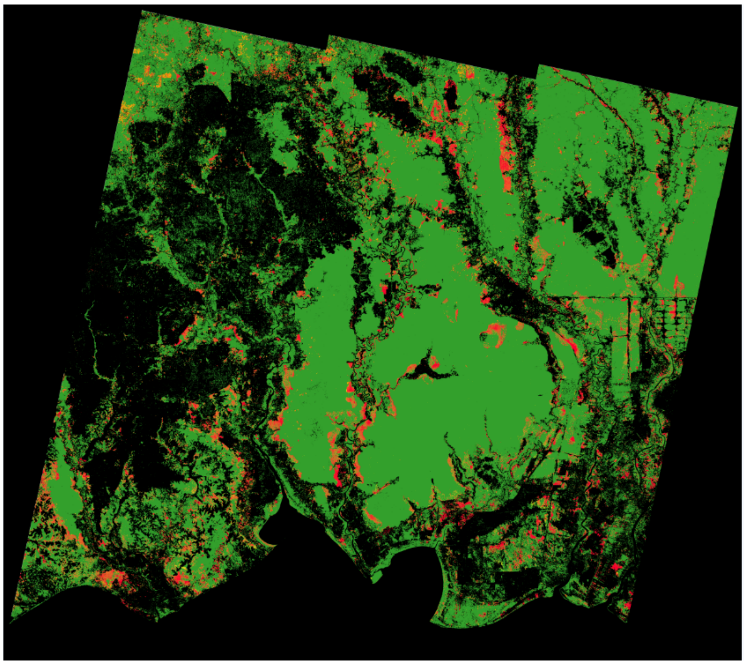


Figure 3.1: Visualization of one large TIFF image.

3.2.3 Area segmentation - Tiles definition

The image resolution (total number of pixels) of each satellite image profoundly affects the computing power requirements to create and use the machine learning model. Nevertheless, a small image size might not contain the necessary information for the model, resulting in weak predictions. With that in mind, the definition of the tile size results in understanding this trade-off.

The tile will represent part of the total area at a single moment in time. For the deforestation use case, it is vital to have in mind that the prediction can be a result of some aspects known to affect deforestation rates. Therefore, it is a good idea to have them included in the same tile as forest areas. Some examples are farms, rivers, coastlines, roads, and protected areas.

With fewer pixels close by and with no transmission of spatial information between tiles, a tile's border will have a lower prediction potential. This means that the model will not have a clear idea of the surroundings of that area, which can result in uninformed predictions. Figure 3.2 displays a graphic representation of this phenomenon.

To increase this low prediction potential, some mathematical models transmit the information between tiles. For example, using pre-processing techniques by adding a pixel's distance to the point of interest. The points of interest, in this case, are the aspects that affect deforestation rates such as rivers or roads. To capture relevant data, it is beneficial that the team responsible for implementing the solution needs to perform a

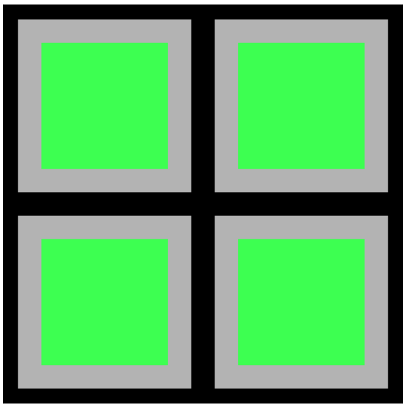


Figure 3.2: Graphic representation of the low border prediction potential phenomenon using four adjacent tiles. Green areas represent pixels with enough neighbors. The border area in grey has lower prediction potential.

geographical and sociopolitical study in every region. This can slow down the EWS expansion in different areas.

Another possibility is the overlap of tile areas, to increase the number of total green areas. This would result in more tiles to process, increasing the training time. Figure 3.3 expresses this idea.

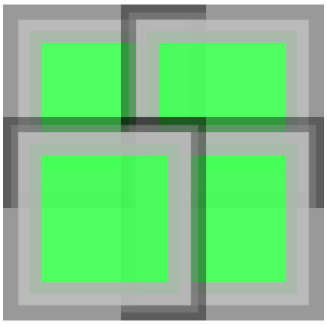


Figure 3.3: Overlapping tiles to increase prediction potential.

3.3 Spatiotemporal sequence forecasting

As the name suggests, spatiotemporal sequence forecasting (STSF) tasks are high-dimensional problems that intersect spatial and temporal dimensions to predict territorial shifts in future steps. Therefore, this is a suitable concept for the deforestation prediction task. Many variables can alter the deforestation probability of a given area. For high-dimensional problems containing rich data sources, statistical and ML techniques are commonly implemented, to account for the conditional weight of multiple features. Taking into consideration unpredictable human actions, Shi and Yeung (2018) mentions that both probabilistic forecasting and stochastic layers can result in better predictions, avoiding blurred images.

3.3.1 STSF Categorization

The recent work of Shi and Yeung (2018) lists and explains current machine learning and deep learning-based methods for STSF based on the state of the coordinates (changing, fixed regular or fixed irregular) and measurements (fixed/changing or changing). Still, according to Shi and Yeung (2018), our satellite image prediction task falls into the Spatiotemporal forecasting on the regular grid category (STSF-RG). Therefore, only the classical machine learning and deep learning methods evaluated in this classification will be taken into consideration when building this framework.

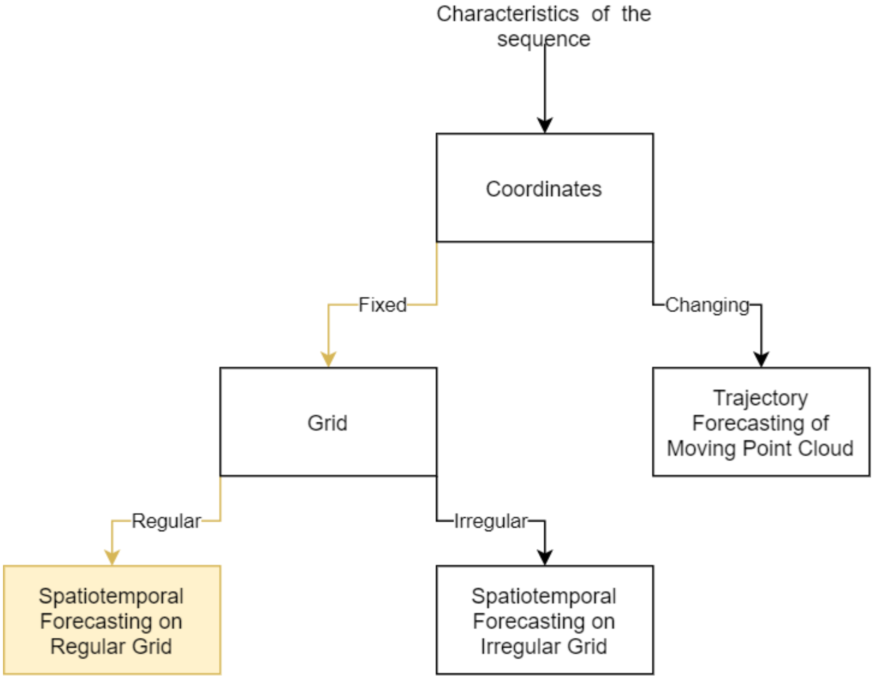

Characteristics of the sequence
Coordinates
Fixed
Changing
Grid
Trajectory Forecasting of Moving Point Cloud
Regular
Irregular
Spatiotemporal Forecasting on Regular Grid
Spatiotemporal Forecasting on Irregular Grid

FIGURE 3.4: Process to categorize a spatiotemporal sequence forecasting

3.3.2 Possible algorithms

Based on our assumptions and conclusions so far, Table 3.1 displays DL methods reviewed by Shi and Yeung (2018) that are suitable for STSF-RG tasks and deals with the deforestation case uncertainties. The categories are divided into classical ML with subcategories State Space Model (SSM) and Gaussian Process (GP), and Deep Learning (DL) with subcategories Deep Temporal Generative Models (DTGM) and FeedForward Neural Networks (FNN) & Recurrent Neural Networks (RNN). With this shortlist of methods in hand, we can check individually if the subcategories have limitations that require extra pre-processing or restrain their usage in the deforestation prediction use case.

Abreviation	Method name
TSBN	Temporal Sigmoid Belief Network
SRTRBM	Structured Recurrent Temporal Restricted Boltzmann Machine
FC-LSTM	Fully-Connected Long Short-Term Memory
3D CNN	3D Convolutional Neural Networks
VAE	Variational Auto-encoder
GANs	Generative Adversary Networks
DC-CNN	Dual Channel Convolutional Neural Networks

Table 3.1: DL methods that deal with uncertainties

The methods found in both DTGM and FNN & RNN subcategories are based mostly in Structured Recurrent Temporal Restricted Boltzmann Machine (SRTRBM), Temporal Sigmoid Belief Network (TSBN), 3D Convolutional Neural Networks (3D CNN), Fully-Connected Long Short-Term Memory (FC-LSTM), Generative Adversary Networks (GANs), and Variational Auto-encoder (VAE).

Dual Channel Convolutional Neural Networks (DC-CNN) usability for change detection in satellite images (Liu et al., 2017) indicates that a similar approach can predict deforestation. Nevertheless, it is essential to clarify that today there is no publication generating a model with this goal.

3.4 Generating interpretable models

To interpret a DL model with image inputs, class activation maps (CAM) are a recurrent method in the literature with positive results in different applications (Kwaśniewska et al., 2017, Meng et al., 2019, Yang et al., 2018, Zhang et al., 2018). It highlights the most critical factors that induce a prediction. Figure X displays a CAM example interpreting the classification of X.

According to Pope et al. (2019), the implementation of CAM requires the layer before the output layer to be a convolutional layer followed by a global average pooling. This confirms experiments done by Yang et al. (2018). His work also displays methods to do a 3D CAM on 3D convolutions, which can be useful to capture the explanation among the height, width, and time dimensions.

3.5 Predictions evaluation

Rainforests, such as the one in Borneo, are well known for their large areas and by the difficulty of navigation. That said, not all future predictions will be able to be confirmed

		Ground Truth	
		Positive ground truth	Negative ground truth
Prediction	Positive prediction	True positive (TP)	False positive (FN)
	Negative prediction	False negative (FN)	True Negative (TN)

Table 3.2: Confusion matrix building blocks

by ground agents. Therefore, it is beneficial that the model assessment consists of values that elucidate the prediction usability. For this work, the ground truth will be the only satellite collected data that will be used to evaluate the ML models.

3.5.1 Evaluation metrics

Pixel-wise, this STSF tries to predict the pixel classification. When comparing predicted classes with the ground truth, a natural result is a confusion matrix. The confusion matrix describes the model performance by analyzing true positives (TP), true negatives (TN), false positives (FP), and false negatives (FN). Those values can be combined in order to evaluate the model. Table 3.2 displays a generic confusion matrix. Table 3.3 exemplifies the most common evaluation metrics that can be extracted from it.

As most real-life scenarios (Hong et al., 2007), the distribution of the raw data is not uniform among the classes. This imbalanced data set imposes a challenge when evaluating the generated models. Two possible solutions are applicable in model training techniques (Hong et al., 2007) or data re-sampling. There is no universally right choice of how to re-sample data sets (Burnaev et al., 2015). Since this thesis main objective is to build a guide to predict deforestation using DL and satellite images, re-sampling adds a critical layer of specificity that can lead to low evaluation quality.

A third possible solution is to prioritize more informative evaluation metrics for unbalanced datasets and ignore metrics negatively affected by skewed data distribution. One example is accuracy, it measures how often the classifier is correct, but since it gives the same importance to TP and TN, it can be misleading on imbalanced datasets. As a replacement, Brodersen et al. (2010) proposes the usage of balanced accuracy, which is the sum of the recall with specificity divided by two, as seen on equation 3.1. Saito and Rehmsmeier (2015) carried out an extensive analysis of the most commonly used evaluation metrics. Based on this work, a Precision-Recall (PRC) plot better represent the model performance on an imbalanced dataset.

Evaluation metric	Formula
Accuracy	(TP+TN)/(TP+TN+FP+FN)
Precision	TP/(TP+FP)
Recall	TP/(TP+FN)
Specificity	TN/(TN+FP)
Fall-out	FP/(FP+TN)
False omission rate	FN/(FN+TN)
False discovery rate	FP/(FP+TP)
F-score	2 * (Precision * Recall)/(Precision + Recall)

Table 3.3: Common evaluation metrics extracted from the confusion matrix

3.5.2 Evaluation process

To evaluate the different models, the same inputs will be provided to the models. Based on the prediction, two binary confusion matrix will be built for each critical class prediction (degradation and deforestation). This matrix will result in a comparative report based on each model's performance on precision, recall, and balanced accuracy according to equation 3.1.

$$BalancedAccuracy = \frac{Recall + Specificity}{2} \tag{3.1}$$

Part IV: Practise

Chapter 4

Artificial Neural Network Architectures

This chapter presents the construction of some of the machine learning architectures that can be used for the satellite prediction use case based on the previous algorithms and the literature.

4.1 Inputs and outputs

To increase the usability and replicability of the solution, the NN models will accept as an input a sequence of five squares RGB image tile. They will provide as an output one image with the same format. As a proof of concept, this work will focus on producing predictions only of one time-step ahead of the last know image.

4.2 Conv3D

Building upon the knowledge explained in section 2.2.2, a 3D convolutional NN uses that principle with a 3D filter. After stacking the sequences of inputs together, the convolution can happen on the dimensions of height, width, and time, capturing the evolution of deforestation with time. Figure 4.1 displays this process.

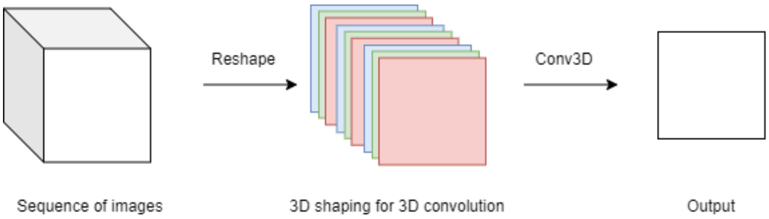


FIGURE 4.1: High level Conv3D input to output process.

4.3 ConvLSTM + Conv2D

The ConvLSTM, as explained in section 2.2.3, combines the gating of LSTM with 2D convolutions. To have a similar output and to gather more spatial awareness, a 2D convolutional layer is added after the ConvLSTM. According to the experiments done by Thangarajah (2019), a ConvLSTM has more trainable parameters, increasing the required computational power. Two architectures based on this were created, one using one ConvLSTM layer and another with two adjacent layers. This can be seen on figures 4.2 and 4.3

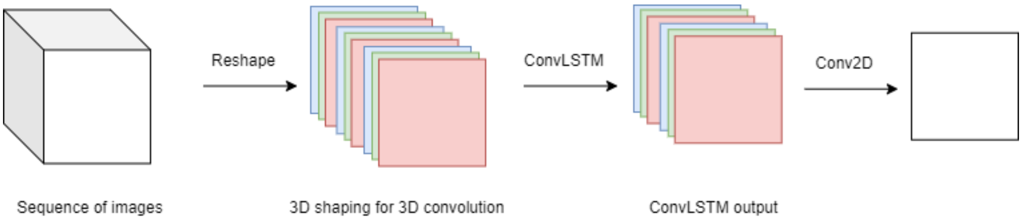


FIGURE 4.2: High level ConvLSTM + Conv2D input to output process.

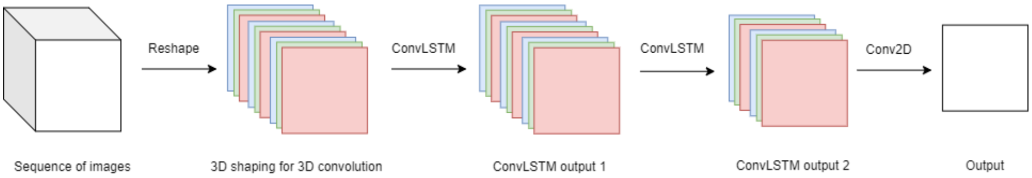


FIGURE 4.3: High level double ConvLSTM + Conv2D input to output process.

4.4 GANs - LSTM

This architecture is a combination of GANs, CNN, and LSTM. Xu et al. (2019) proposes to use an LSTM to obtain fixed dimensional vector representation. This vector will be the input of a previously trained generator that will generate the final image. An overview of this idea can be seen on 4.4.

The LSTM use is due to the power to understand the progressive change. The deforestation evolution of a sequence of images is transformed into the fixed dimensional vector

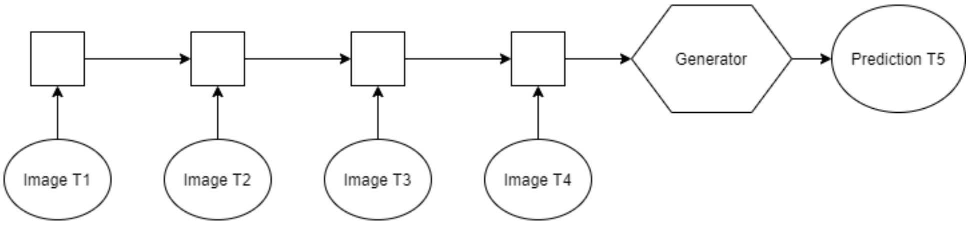


Figure 4.4: Graphic representation of GANs-LSTM architecture. The details of the LSTM can be found on chapter 4.

representation, which will serve as the input for the generator. As seen in appendix A, GANs are trained using a Generator and a Discriminator. After training a generator using input a vector of the same size used on the LSTM, the generator is attached to the GANs-LSTM architecture.

Between the many variations of GANs, currently, the one with better results on similar use cases was proposed by Radford et al. (2015). The Deep Convolutional GANs (DCGANs) replaces pooling layers with stridden or fractional-stridden convolutions. Another part of the DCGAN internal design is the batch normalization. This means that the training performs a normalization for each training batch: batch normalization to have faster training, better regularization, and accuracy (Ioffe and Szegedy, 2015). As for activation function, the best use is the rectified linear activation (ReLU) (Nair and Hinton, 2010) for the generator layers except for the output that uses Tanh. All the discriminator layers use LeakyReLU as activation function, instead of the maxout used by Goodfellow et al. (2014).

Chapter 5

Experiments

To experiment with the proposed architectures, this work performs a comparative study using the same database. The database consists of 6 big areas captured on TIFF files throughout 85 to 105 different time steps. The earliest image was generated displaying the deforestation from September of 2015 to October of the same year. The most prolonged time interval is from September of 2015 to December of 2019. Each of the six large areas has different dimensions, but all are around 20.000 pixels. To process all of it at once, the hardware requirements are out of this research's reach. Therefore, preprocessing steps are required.

5.1 Experiment specifications

The programming language used in this experiment is Python due to the large available neural networks related frameworks and online community. The processing is partially done by a GeForce GTX 1050Ti, where the CUDA Toolkit from NVIDIA is essential to speed the training process. The computer that performed the training also has 16GB of RAM, an Intel i7 2.80GHz CPU, and an SSD with 20GB.

5.2 Tile segmentation

Ideally, the TIFF files should be treated as raster data files. Doing so results in maintaining details such as coordinates and class values. However, it also impacts on the visual representation of the output. To maintain a similar aspect on the output, the TIFF files will be treated only as regular images.

As previously mentioned in section 3.2, when cropping a tile into smaller tiles, it is essential to overlap information to diminish the effect of fewer neighbors pixels across images. Also, since this training objective is to understand Spatio-temporal relations on the images, a larger image contains more possibilities of understanding the right correlations. Because of computer power limitations, three different sizes were used on the experiments, 256, 512, and 720 squared pixels. With a smaller tile, fewer pixels are predicted and less information to help the prediction; for those reasons, the 256 squared pixels experiments were discontinued. Figure 5.1 displays an example of how it is more likely to have changes on bigger tiles.

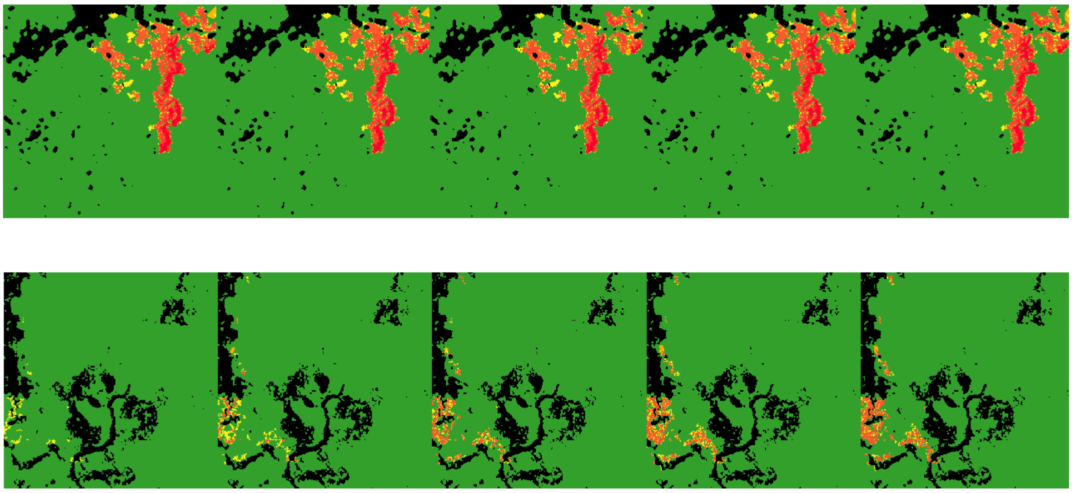


Figure 5.1: Sequence of tiles representing the evolution of deforestation. On top images with 256 squared pixels. At the bottom images with 720 squared pixels.

An automatic script to crop the TIFF files into smaller tiles is responsible for the job. There, a unique height and width define the smaller tiles. To generate a more significant number of smaller tiles and mitigate the issues of tile segmentation, the crop process produced overlap among neighboring tiles on half of the dimension that connects them.

Since the large TIFF files contain large areas with no data (only black pixels), before saving the smaller tiles, the script checks if the tiles have different pixel colors. This avoids adding data that will not enrich the model training. To maintain the data organized, a folder is created for each smaller tile, containing its evolution through time.

To avoid overfitting, an algorithm randomly sort the folders into train, validation and test sets. The training set will be used to fit the model, the validation set to tune the models' hyperparameters, and the test set will be used for evaluation. The Pareto principle (Dunford et al., 2014) indicates a general understanding that for many events 80% of consequences are produced by 20 % of causes. To try to maintain a similar ratio, the proportion used is 75-20-5. Table 5.1 displays the total number of tiles for each set.

Tile Size	Train Tiles	Validation Tiles	Test Tiles
720	161 000	32 400	12 100
512	210 700	41 000	14 700

TABLE 5.1: Total number of tiles for each tile size segmentation.

5.3 Batch generator

To process this amount of data, a batch generator is responsible for fetching it from the folders and feeding it to the model training process. The generator receives as input the batch size and the main image directory and, after a directory shuffle (therefore tile shuffle), enter the same number of folders respecting the batch size. Once inside the folders, it will randomly start from one of the 80 first steps and gather a sequence of six images, were the first five are to use as training input and the sixth to compare with the prediction. By randomizing both folders and images, the training process accesses different areas and periods for each batch. Layers containing LSTM cells would better benefit longer sequences, but this also increases hardware requirements.

5.4 Models training

5.4.1 Loss function

The experiments use the RGB values, therefore the training uses a regression loss. One of the most common regression loss is the mean squared error. Its main characteristic, is penalizing errors that are far from the true values more in comparison to errors close to the true values. This can help to penalize big mistakes such as having a red pixel being green or vice versa.

5.4.2 Kernels

Since all the models were based on CNN, at least one of the layers has a kernel. Each kernel results in a number of parameters based on its dimensions. The parameters number calculations and layer positions can be found in Appendix C.

5.4.3 Hyperparameters

The hyperparameters choices need to take in consideration the hardware limitation, the data distribution and the availability on the most used ML frameworks. Based on this,

id	Architecture	Tile size
0	ConvLSTM with gaussian noise	512
1	ConvLSTM without gaussian noise	512
2	Double ConvLSTM without gaussian noise	512
3	Conv3D	512
4	Conv3D	720

Table 5.2: Models trained to predict next image

id	Architecture	Tile size
5	ConvLSTM with gaussian noise	512
6	Conv3D	512

Table 5.3: Models trained to predict last image

Adam (Kingma and Ba, 2014) is the chosen optimizer. It is a stochastic gradient descent method with adaptive learning rates from estimates of first and second moments of the gradients. The chosen values of learning rate, beta 1, beta 2 and epsilon are the current defaults used in the Keras implementation (0.001, 0.9, 0.999 and 1e-7 respectively). After testing with different epoch sizes and without a lot of improvement on the loss function after the 30 epoch, the number of epochs is established in 30.

5.4.4 Models trained

The hardware available to perform the experiments limited the possibilities. The GANs-LSTM architecture required more computing power for the selected tile sizes. For the tile size 720 squared pixels, the only possible architecture was a single conv3D layer. For 520 squared pixels, the possible options were a single conv3D layer, convLSTM + conv2D (with and without Gaussian noise), and double convLSTM + conv2D. The Gaussian noise is used to reduce overfitting, therefore increases the model generalization. For the experiments that use Gaussian noise, this is done by using a regularization layer (therefore is only active during training) with the standard deviation of the noise distribution fixed at 0.1. Table 5.2 displays the models trained and Appendix C the scheme of each model.

In order to validate and compare these models, two more models were trained to predict the last time step (so no change). The list and architectures are discussed in Table 5.3.

By architectural design, the model outputs are RGB images with the same dimensions as one of the input images. Figure 5.2 compares a tile ground truth with the Conv3D output image. The yellow circles highlight the pixel change color around a degradation area that is increasing with time. The blue circle points to an area in which the prediction was not correct. Figure 5.3 displays the train loss while Figure 5.4 displays the validation loss.

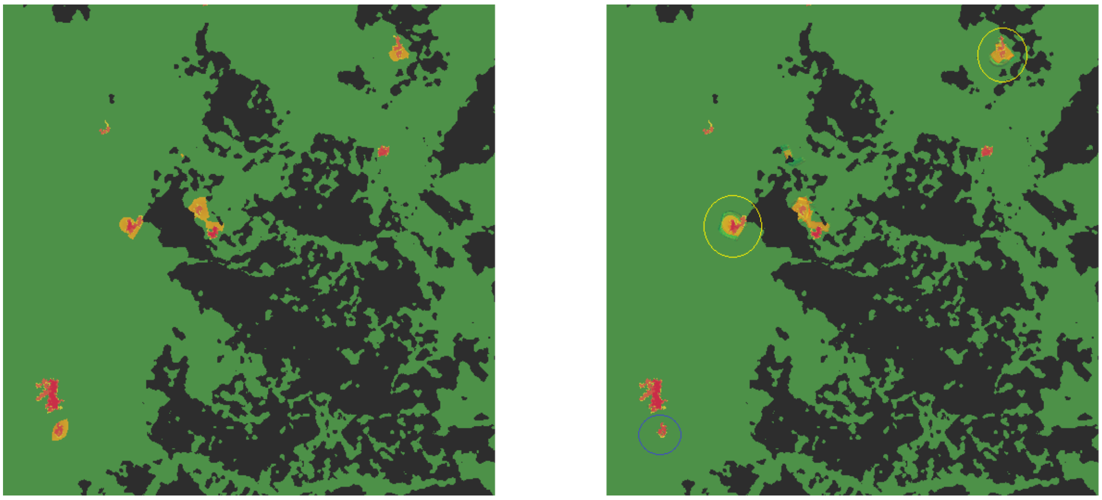


FIGURE 5.2: Ground truth (left) and prediction (right) of the Conv3D model for 512 squared pixels.

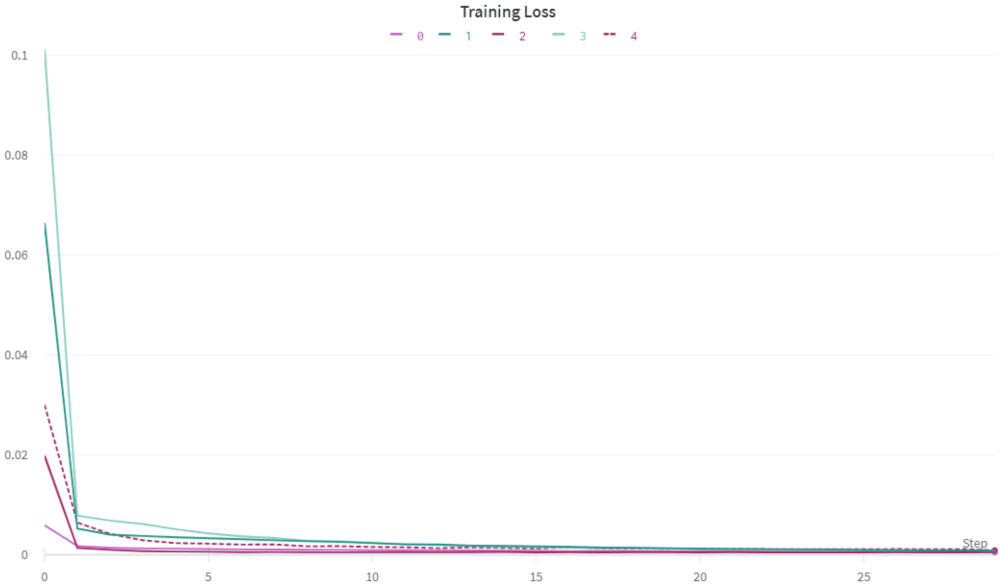


FIGURE 5.3: Train learning curve.

Since accuracy is not a good metric for unbalanced data sets, the experiment uses *perceptual distance* is the other metric taken during training and validation. The figures 5.5 and 5.6 displays the perceptual distance metric evolution while training and validating. This metric is based in a comparison of two images using each one of the RBG bands. The code used for this metric can be found in Appendix B.

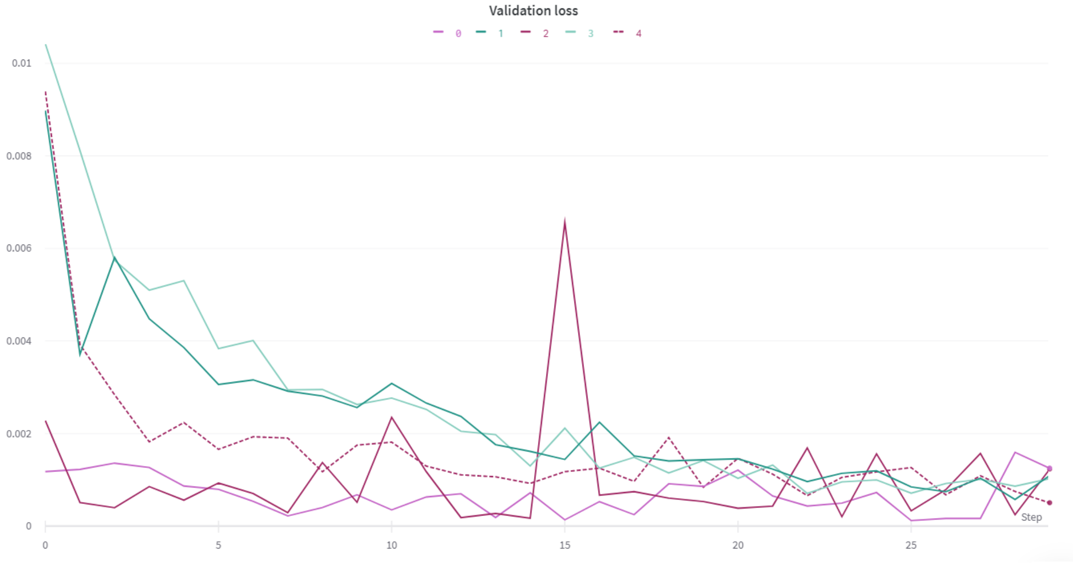


FIGURE 5.4: Validation learning curve.

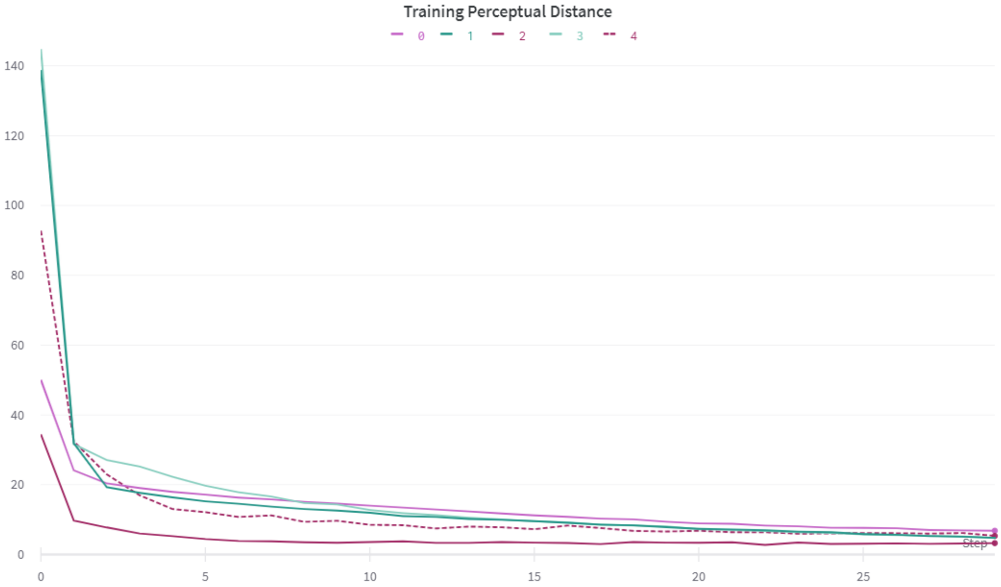


FIGURE 5.5: Train perceptual distance.

5.5 Model Evaluation

Once the models are generated, they are evaluated by creating predictions using the test data and comparing the result with the next image. The algorithm found in Appendix B builds boolean matrixes for each prediction-real pair about the two most important classes: deforestation and degradation. A spectrum of the image's red and green bands are used to capture the pixel color.

The evaluation process had a limit of 100 images. This is due computational issues and

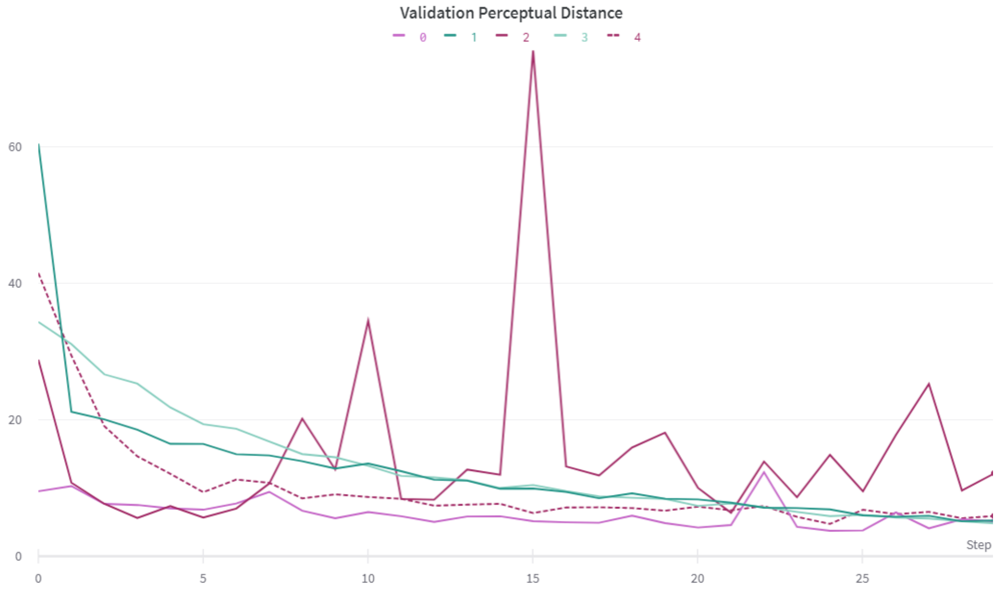


FIGURE 5.6: Validation perceptual distance.

to have a similar number of examples when evaluating the models. To validate this process, the evaluation was performed on two random splits of the test data, resulting in similar results. Based on the information of the chapter 3, the next step is comparing the precision and recall of all the models via two scatter-plots, as seen on figure 5.7 and 5.8. Another evaluation metric mentioned in the same chapter is of balanced accuracy. All the models' balanced accuracy and F1 scores are on table 5.4.

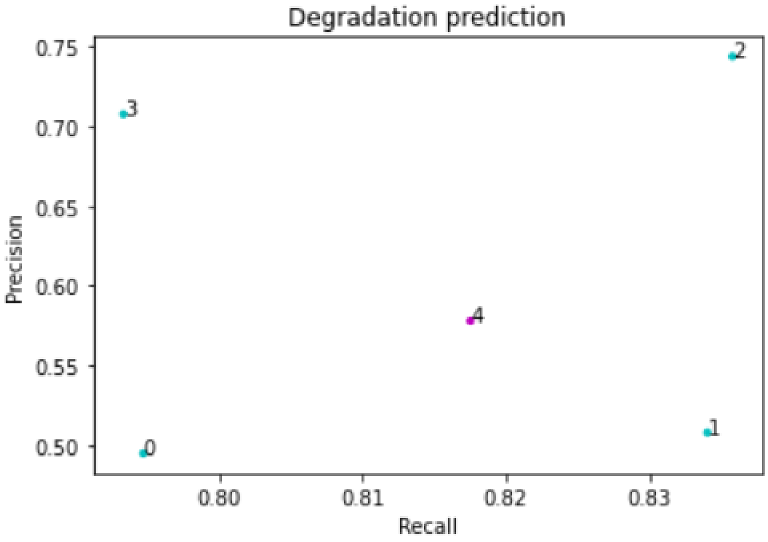


FIGURE 5.7: Precision and recall of each model for degradation prediction.

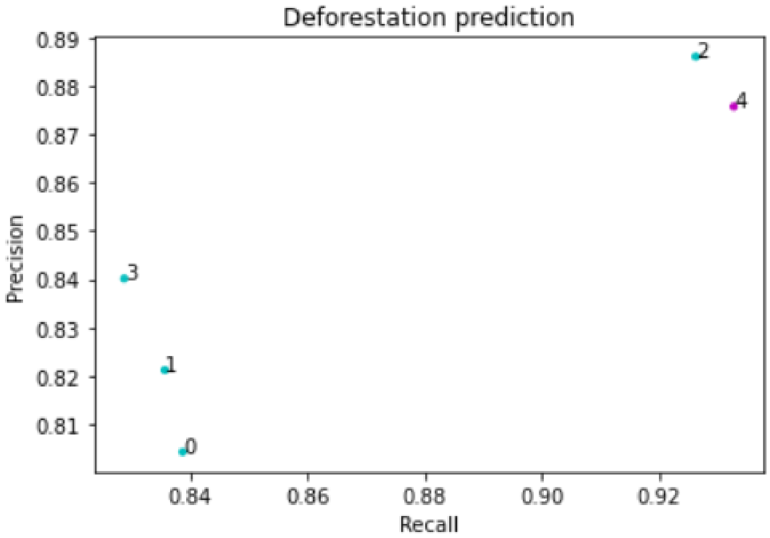


Figure 5.8: Precision and recall of each model for deforestation prediction.

	Degradation		Deforestation	
id	Balanced Accuracy	F1	Balanced Accuracy	F1
0	0.880	0.610	0.890	0.821
1	0.908	0.631	0.8914	0.828
2	0.907	0.787	0.945	0.906
3	0.890	0.748	0.893	0.834
4	0.901	0.677	0.943	0.903
5	0.989	0.785	0.957	0.913
6	0.996	0.954	0.960	0.953

Table 5.4: Balanced Accuracy and F1 score rounded to the thousandths place

5.6 Discussion

When predicting degradation within the 520 squared pixel models, the lack of Gaussian noise contributed to higher recall and precision. Using an extra layer of ConvLSTM also resulted in the best precision for both degradation and deforestation.

When comparing different tile sizes, the degradation prediction of the 720 squared pixel model had a higher recall but worse precision compared to its 520 squared pixel counterpart. On deforestation prediction, both metrics increased.

An impressive result was the low performance on precision for the models with a ConvLSTM layer followed by a Conv2D. The Conv3D layer seems like a better fit to capture the three-dimensional dependencies. This changes if we add one more ConvLSTM layer, but the computational costs increase, so for each, a trade-off analysis is vital before deciding which path to follow.

As expected, after analyzing the scatter plots, the double ConvLSTM architecture has the best-balanced accuracy for deforestation and a close second best for degradation. This, as well as its position on the prediction-recall scatter plot, indicates that this architecture is the most successful architectures tested.

All the evaluation of the models 5 and 6 (the models that predict the last time step) are compared with the evaluation of models 0 and 3. This is because they share a similar structure. As expected the values of Balanced Accuracy, F1, precision and recall are higher in the models 5 and 6. A plot comparing the precision and recall values can be found in Figures 5.9 and 5.10. It is notable that the precision metric on the model 5 is better than 0, but still worse than the different architecture predicting the next time step.

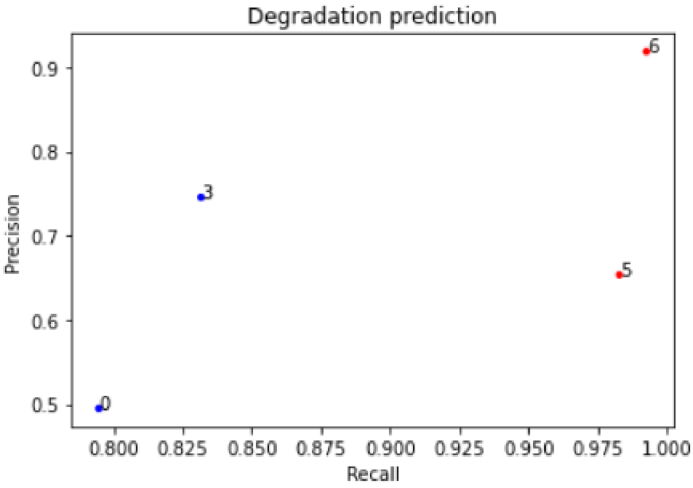


FIGURE 5.9: Precision and recall of each model for degradation prediction. Comparison of models with similar architecture but different goals. In blue predicting the future time-step. In red predicting the last time-step.

The models 0 to 4 need to be better than using the last step before validation as the next step prediction. As a final validation, this solution will have the id 7.

As seen on Figures 5.11, 5.12, and table 5.5 the option of just using the last time step gives better results. Two main things explains this. First, the tile sizes might be so small that is not capturing a lot of differences for each time step. Furthermore, the passage of time between selected tiles results in just a little change, so small that is hard to predict in just one more time step. During the next chapter some possible solutions are mentioned in the future work section.

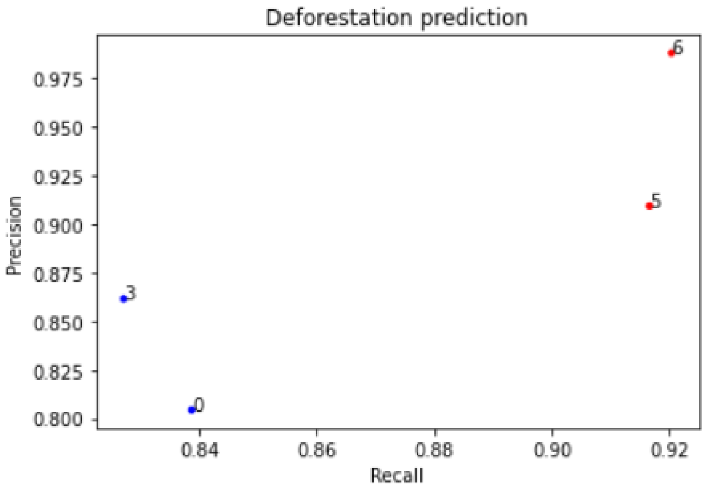


FIGURE 5.10: Precision and recall of each model for deforestation prediction. Comparison of models with similar architecture but different goals. In blue predicting the future time-step. In red predicting the last time-step.

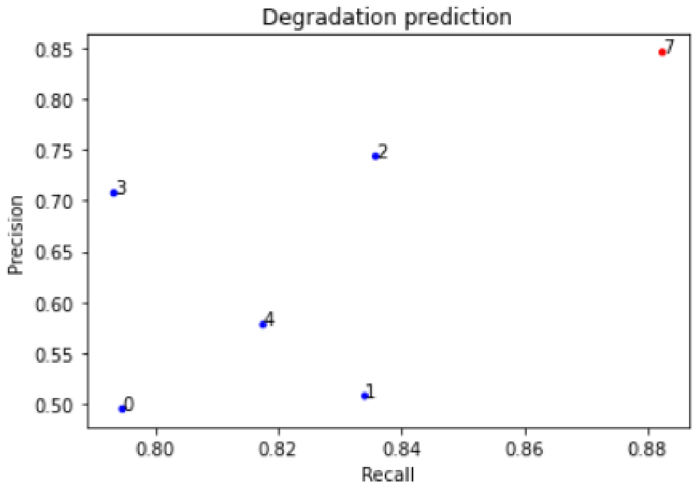


FIGURE 5.11: Precision and recall of each model for degradation prediction. Comparison of models with similar architecture but different goals. In blue predicting the future time-step. In red predicting the last time-step.

	Degradation		Deforestation	
id	Balanced Accuracy	F1	Balanced Accuracy	F1
0	0.880	0.610	0.890	0.821
1	0.908	0.631	0.8914	0.828
2	0.907	0.787	0.945	0.906
3	0.890	0.748	0.893	0.834
4	0.901	0.677	0.943	0.903
7	0.930	0.864	0.980	0.967

TABLE 5.5: Balanced Accuracy and F1 score rounded to the thousandths place

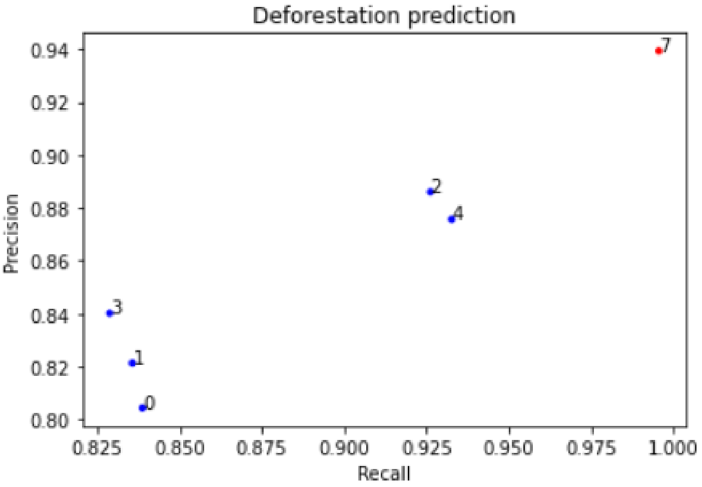


Figure 5.12: Precision and recall of each model for deforestation prediction. Comparison of models 0 to 4 (in blue) with using the last time step as prediction (in red).

Part V: Results

Chapter 6

Discussion

Science, by its nature, always expands and creates new possibilities. It is imperative that advances in machine learning techniques are applied to create a good impact on the world. Forest loss impacts everything since it can result in climate change, loss of biomes, and a decrease in the total number of species on this planet. This chapter summarizes the findings of this work and relates them to the research question proposed in chapter 1.

6.1 Experiments discussion

Originally, the large TIFF files have a limited number of colors for deforestation and degradation. When using RGB bands to train and predict the future time steps, the results had a larger color range. This is good for human visualization, since the borders between different areas are more soft if compared to the large TIFF files. But, as a trade-off, only regression loss functions can be used, excluding the possibility of using classification loss functions.

While doing manual model inference, some patterns emerged. First, it was much easier for the models to predict the transition from degradation to deforestation than from forest to degradation (or deforestation). Second, classification changes next to black pixels were also more likely to be predicted (as displayed on figure 5.2). Additionally, predictions of the pixels next to the tile borders were not successful. Futhermore, the areas with predicted changes often present a shadow-like aspect, as seen in Figure 6.1. This is probably due to a confusion when predicting the RGB bands.

As expected, because of its structure, the ConvLSTM models contains a significant higher amount of parameters than Conv3D models. Each layer of ConvLSTM adds around 2000 parameters, while one layer of the Conv3D only adds 136 parameters. This means that the computational power required for training and inference of the Conv3D models is lower.

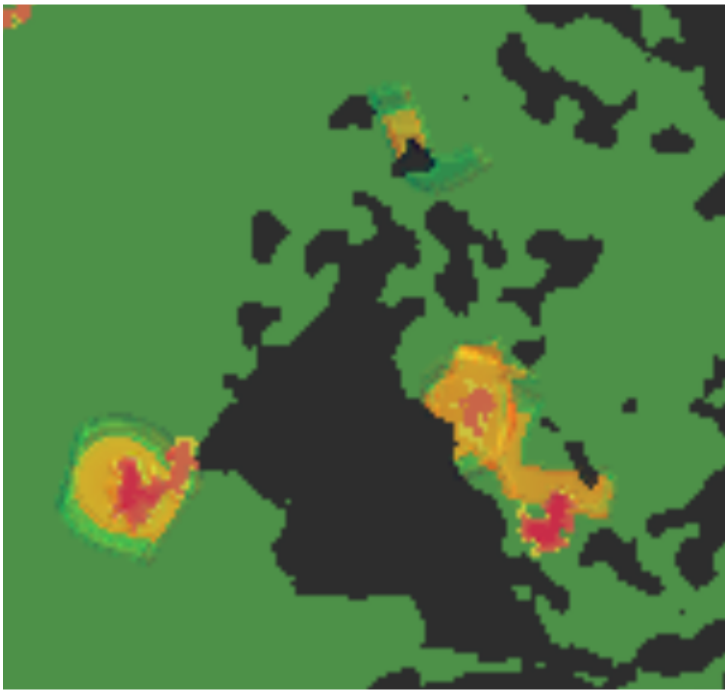


FIGURE 6.1: Prediction areas with shadow-like aspect.

6.2 Sub-questions

6.2.1 What are the current deep learning architectures and methods suitable to predict deforestation with satellite image data?

First, to answer this question, it is important to define which generic terminology better describes this use case. As a high-dimensional sequence prediction problem, the terminology ”spatiotemporal sequence forecasting” is used in the literature to similar group tasks. With fixed geographic coordinates and regular grid, the suitable deep learning architectures are mentioned in table 3.1. The chosen methods also deal with the tasks uncertainties. Some of them are explained in appendix A and 4, from the building blocks and background theory to internal architecture.

6.2.2 What are the design choices to choose and combine techniques to create the deep learning architectures for this use case?

When working with high volumes of data, it is imperative to first understand what the data represents and its format. The representation of large areas is perceived differently depending on the coordinate system. To avoid adaptation issues, the chosen coordinate system need to be unique across the dataset and code. A geographic system is more suitable for large areas because it uses degrees to capture the three-dimension surface of the world. Because of the earth's shape, the higher the distance between a chosen area and the equator, more distortions are seen on projected systems. In summary, geographic coordinate systems are suitable to point where and projected coordinate systems to say how to draw on a surface.

The next point is the data itself. There are available data on the internet, but it is not easy, or free, to gather sequences of large areas with a short period between each image. Radar technology and classification algorithms can aid in creating a more precise vision of forest areas.

As discussed in chapter 6, the selected architecture needs to respect the available computational power. One possible workaround to run deeper and more complex models is the definition of the tile area. Subsection 3.2.3 explains a possible way to crop larger tiles and how the prediction potential is generally lower in the border of tiles.

6.2.3 Which techniques should be used to facilitate the interpretability of the model's results?

The usual black box of a DL model opens with a class activation map. A CAM clarifies the critical classes for each classification. The goal of deforestation prediction is to lower deforestation, easy to understand interpretable results facilitating the connection between the provided correlation to case-specific causation.

6.2.4 What are the most suitable performance measures for this domain?

Deforestation or any other satellite-based prediction will count with a skewed distribution of data. Although the field of statistics created tools to deal with this, it can lead to biased models. To avoid it and still have valid evaluation metrics, the metrics with less or no impact of true negatives are more suitable to measure performance. Therefore balanced accuracy, precision, recall, and the F1 score better demonstrate how good the model is.

6.3 Main research question

> How can a framework be developed to aid data scientists in selecting deep learning techniques and process satellite data towards providing accurate and interpretable short-term predictions?

This work assembles different areas of knowledge required to create a good project on interpretable deforestation prediction using satellite images. For this, possible pitfalls, design choices, ML algorithms and evaluation metrics are presented throughout the thesis. Although most of the DL architecures presented during Chapter 3 were not possible to be validated because of hardware requirements, the experiments with simpler models already had good results among the evaluation metrics.

6.4 Limitations

Most of the selected DL techniques require hardware that is not easily available. Therefore the experiments chapter is not as developed as I first envisioned. The experiments with simpler models already indicate that this technology is a good option.

Because of my lack of experience when working with satellite data, part of the research time was employed on a dual channel DL implementation that was not completed. The current EWS project by Deloitte has the radar data of Borneo, but it does not have the raw satellite data. A DL model can learn new patterns with this raw satellite data, creating a more generic final product.

The initial planing for this thesis was also experimenting with CAM and create a interpretable solution. To apply CAM the model needs a more complex architecture and the hardware limitations did not allow the models training during the execution of this work to be adapted. If the implementation was successful, each pixel choice would create its own CAM that explains its color.

6.5 Future research

6.5.1 Architectures implementation

The natural next step is the implementation of class activation maps to generate interpretable results. Later on, experiment and implement more methods discussed during Chapter 3. This can result in a more detailed implementation process on different architectures and a comparative evaluation process within the same dataset.

6.5.2 Time step research

As explained in the last part of the last chapter, the experiments done resulted in models worse than just using the last image as a prediction. A good next step for this research is to understand how big needs to be the time step so that the model can predict better than just using past images.

6.5.3 Input variation

Changing the input data format from RGB images to raster data should lead to similar results, but this needs to be validated. Once done, this opens possibilities of applying dual channel CNN (Gao et al., 2017, Liu et al., 2017) by combining the TIFF files with raw satellite data.

6.5.4 Cost calculation

The investment in a project to prevent deforestation usually comes from NGOs or the public sector. Cloud services can host the training and usage of a satellite prediction solution. It is beneficial to understand the costs of each step of the process to compare with the potential avoided deforestation.

6.5.5 Model generalization

The experiments were done with data from Indonesia. This same model can be used in a different landscape? How does it perform comparing to a model trained with data from a different country? A comparative study about the model generalization can be interesting for a future EWS implementation.

Appendix A

Artificial Neural Networks

This appendix will discuss artificial neural network concepts that are used and discussed during this work.

A.1 Backpropagation

Backpropagation is a method that, while training a NN, compares its output (a prediction or a classification) to ground truth data, then it calculates the gradient of a given a given error function from the last to the first layer, changing the neural networks weights (Buntine and Weigend, 1991).

A.2 Feedforward Neural Networks

Sandberg et al. (2001) states that Feedforward Neural Networks (FNN) by themselves are nonlinear static networks. A NN that do not contain feedback loops between their layers; there, the outputs of the model are not fed back between layers (Montana and Davis, 1989). This means that the acyclic design of an FNN has a direct contrast with the cyclic RNN (Schmidhuber, 2015). This type of network is usually easier to implement and have broader use (LeCun et al., 2015). This is due its easy generalization and the usage of Backpropagation to tune its weitghts.

A.3 Recurrent Neural Networks

Recurrent Neural Networks (RNN) is a type of NN that contain dynamical memory, therefore it is capable to process temporal factors within the input data (Lukoševičius and Jaeger, 2009). RNNs use the concept of internal state vectors to keep information about the history of past elements (Botvinick and Plaut, 2006). The state vectors affect inputs and future timesteps. Because of the gradient effect of backpropagation, usual RNNs suffer from something called *Short-term memory*, where distant events have less impact on new predictions. In figure A.1 there is a typical example of RNN usage. When learning with the past elements, the state vector storage the past words. In a latter state, the first words have less effect when trying to predict the next element.

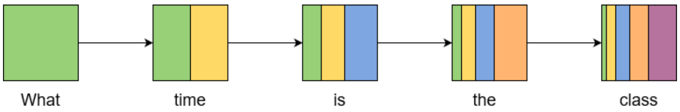


FIGURE A.1: Graphic representation of the state vector analyzing past information to understand current events.

A.4 Restricted Boltzmann Machine

The Restrited Boltzmann Machines (RBM) are a special type of generative neural network, first discussed by Smolensky (1986). This energy-based network consists of two layers (visible and hidden) with undirected connections betweeen them, forming a bipartie graph. The neurons are binary and do not have any conexions within within the same layer. They can be used for supervised or unsupervised machine learning use cases. Figure A.2 is an example of a RBM topology.

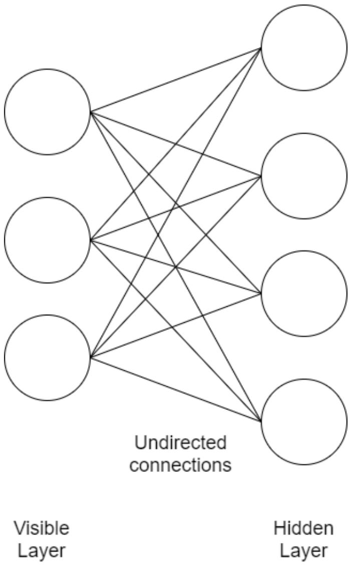


FIGURE A.2: RBM topology example.

The matrix **W**, is the connection between the visible layer **v** and hidden layer **h**. The units might contain bias weigths both in the visible **a** and hidden layer **b**. The energy can be defined using a matrix notation in equation. A.1:

$$E(v,h) = -a^T v - b^T h - v^T W h \tag{A.1}$$

The energy is used to understand the probability distribution. The equation A.2 demonstrates the relation between **E** and **Z**, where **Z** is the partition function that acts as a normalizing factor.

$$P(v,h) = \frac{1}{Z} e^{-E(v,h)} \tag{A.2}$$

A.5 Sigmoid Belief Nets

The Sigmoid Belief Nets (SBN) are directed acyclic locally normalized models composed of binary stochaistic variables (Neal, 1992). They were created and developed to overcome the limitations of back-propagation. To generate data from the model, first the neurons on the top layer, based on their biases, will result in 0's or 1's. The states calculated in the top layer feed the middle layer generating stochastic decisions about what the neurons on the middle layer should be doing. Those results move to the visable effect layer, deciding about what the visible effect should be. Figure A.3 is an example of a SBN topology.

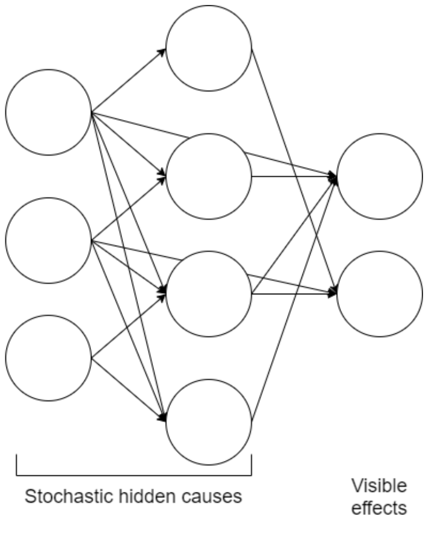


Figure A.3: SBN topology example.

A.6 Deep Belief Network

The Deep Belief Networks (DBN) mix undirected and directed connections between the nodes (Hinton et al., 2006). They usually consist of four layers. The first two are hidden layers with undirected connections, similar to RBMs, and the last two layers have directed connections resembling SBNs. They are a generative graphical model, so they can reconstruct inputs based on examples in an unsupervised way. Figure A.4 is an example of a DBN topology.

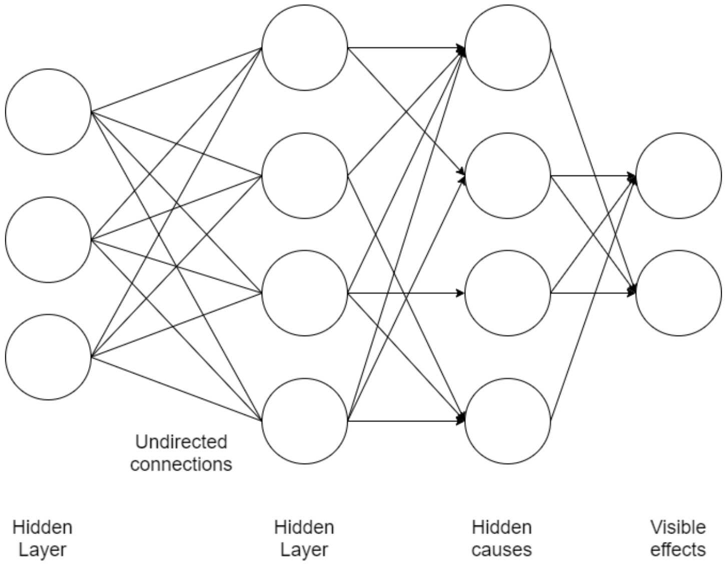


Figure A.4: DBN topology example, the name of the layers are similar as the ones used on the RBM and SBN figures to emphasize their contribution.

A.7 Generative Adversarial Nets (GANs)

Since Goodfellow et al. (2014) proposed the generative adversarial nets framework, the most notable use cases were generating images, video, text, and sounds. Some examples are the high-resolution image synthesis (Wang et al., 2018), bounding box control (Reed et al., 2016), or generating videos with scene dynamics (Vondrick et al., 2016). To understand how they can be used to predict deforestation better, first, this work will explain how they work.

GANs main idea revolves about having two neural networks competing and evolving. The worst-case input for one network is produced by another network. Even though it was not a brand new idea (Schmidhuber, 1992), the difference in the nature of the competition and specification of the learning process popularized GANs, resulting in unique solutions derived from it (Chongxuan et al., 2017, Mirza and Osindero, 2014).

As defined by Goodfellow et al. (2014), given a data set, there are two possible tasks for a generative model. These are density estimation, where we can find the probability density function that describes the data set, or the creation of a function capable of generating more samples that could be seen as part of the initial examples.

The two neural networks that will compete have different roles. One is a generative network (GN), which generates data, while the other is a discriminative network (DN) that exams that data and estimates if it is real or fake. Both neural networks evolve over time, getting better to try to fool (GN to DN) and to detect failed data (DN to GN)

The DN is created by a sample from data. So if a sample x is the source for DN, the goal of DN(x) will be to try to be near 1, so x is a real example. The GN receives a random noise sample z, and the calculations of GN(z) will result in a generated new sample a, which will be evaluated by DN. So while GN will try to make DN(a) be close to 1, DN will try to make DN(a) near 0. The perfect generator would shape the discriminator's DN function to always have as output 0.5 since that would mean it can no longer know what is fake and what is real.

In mathematical terms, both DN and GN behaviour can be explained with this loss function V(DN,GN):

$$min_{GN}max_{DN}V(DN,GN) = E_{x\ pdata(x)}[logDN(x)] + E_{z\ pz(z)}[logDN(GN(z))] \quad \text{(A.3)}$$

The development and increase of the popularity of GANs are mainly because, unlike other ML algorithms, it can tell the model that there can be multiple correct answers. This happens because it doesn't use a unique input-output pair to train and validate the model so that it can mold paths from input to different (and correct) outputs.

A.8 Structured Recurrent Temporal Restricted Boltzmann Machine

The Structured Recurrent Temporal Restricted Boltzmann Machine (SRTRBM) is an extension on the commonly known Restricted Bolztmann Machine (RBM) discussed in the Appendix A. The improvements are found for every word added to the original name. This structure was built over the idea of Temporal Restricted Boltzmann Machine (Sutskever and Hinton, 2007) and Recurrent Temporal Restricted Boltzmann Machine (Sutskever et al., 2009).

Each time-step has its hidden *ht* and visible *vt* vector states. The visible state connects to another hidden state, a recurrent neuron network *rt* that will be an input to *h(t+1)* and *r(t+1)*. Another improvement comparing to a usual RBM is the usage of matrices to mask the connections between the network layers. The model can learn these matrices during the training process in a way to better model the STSF problem. Figure A.5

illustrates the SRTRBM concept, where the undirected W and directed D weights will be part of a dot product with the masking-matrices Mw and Md. Mittelman et al. (2014), who introduced this concept, also performed experiments that display a lower average prediction error for motion capture datasets.

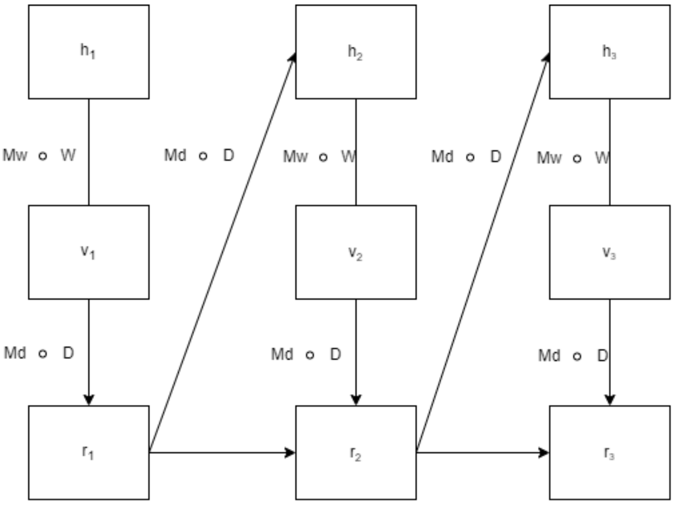


Figure A.5: Graphic representation of SRTRBM.

Appendix B

Model evaluation code

This appendix contain python functions used during the experiments evaluations.

B.1 Degradation evaluation

This function receives as inputs two RGB images and compare them by building a matrix with boolean values. The true values are found when the mix of the red and green bands results in a color between yellow and dark orange.

```
def degradation_evaluation(true_img, pred_img):

    #undo the numbers normalization
    y_true = true_img * 255.
    y_pred = pred_img *255.

    #get red and green bands
    r_true = y_true[:, :, 0]
    g_true = y_true[:, :, 1]
    r_pred = y_pred[:, :, 0]
    g_pred = y_pred[:, :, 1]

    #build boolean matrix
    true_bool = (r_true > 95) & (r_true < 256) & (g_true > 160) & (g_true < 256)
    pred_bool = (r_pred > 95) & (r_pred < 256) & (g_pred > 160) & (g_pred < 256)

    #create confusion matrix
    df_confusion = confusion_matrix(true_bool.flatten(), pred_bool.flatten())

    return calculate_metrics(df_confusion)
```

B.2 Deforestation evaluation

Similar code to degradation evaluation, but changing the values of red and green bands to entail only colors from dark orange to red.

```
def deforestation_evaluation(true_img, pred_img):

    #undo the numbers normalization
    y_true = true_img * 255.
    y_pred = pred_img *255.

    #get red and green bands
    r_true = y_true[:, :, 0]
    g_true = y_true[:, :, 1]
    r_pred = y_pred[:, :, 0]
    g_pred = y_pred[:, :, 1]

    #build boolean matrix
    true_bool = (r_true > 145) & (r_true < 256) & (g_true > 0) & (g_true < 155)
    pred_bool = (r_pred > 145) & (r_pred < 256) & (g_pred > 0) & (g_pred < 155)

    #create confusion matrix
    df_confusion = confusion_matrix(true_bool.flatten(), pred_bool.flatten())

    return calculate_metrics(df_confusion)
```

B.3 Calculate metrics

The confusion matrix calculated during the past functions are the input for this calculate metrics function. The position of each element is the opposite of the explanation found on Table 3.3.

```
def calculate_metrics(confusion_matrix):
    TP = confusion_matrix[1][1]
    TN = confusion_matrix[0][0]
    FP = confusion_matrix[1][0]
    FN = confusion_matrix[0][1]
    precision = TP / (TP+FP)
    false_discovery_rate = FP / (FP+TN)
    recall = TP / (TP+FN)
    specificity = TN / (TN+FP)
    balanced_accuracy = (recall + specificity)/2

    return  balanced_accuracy, precision, recall, specificity, false_discovery_rate
```

B.4 Perceptual Distance

The following function runs with training and validation data. It measures the perceptual similarity between two images by comparing each one of the RGB bands on each image.

```
def perceptual_distance(y_true, y_pred):
    y_true *= 255.
    y_pred *= 255.
    print(y_true)
    print(y_pred)
    rmean = (y_true[:, :, :, 0] + y_pred[:, :, :, 0]) / 2
    r = y_true[:, :, :, 0] - y_pred[:, :, :, 0]
    g = y_true[:, :, :, 1] - y_pred[:, :, :, 1]
    b = y_true[:, :, :, 2] - y_pred[:, :, :, 2]

   return K.mean(K.sqrt((((512+rmean)*r*r)/256) + 4*g*g + (((767-rmean)*b*b)/256)))
```

Appendix C

Model schemes

This appendix contain schemes for the models presented during chapter 5. The number of parameters of Conv3D layers are calculated using Equation C.1 and ConvLSTM using Equation C.2 each layer is calculated as follows:

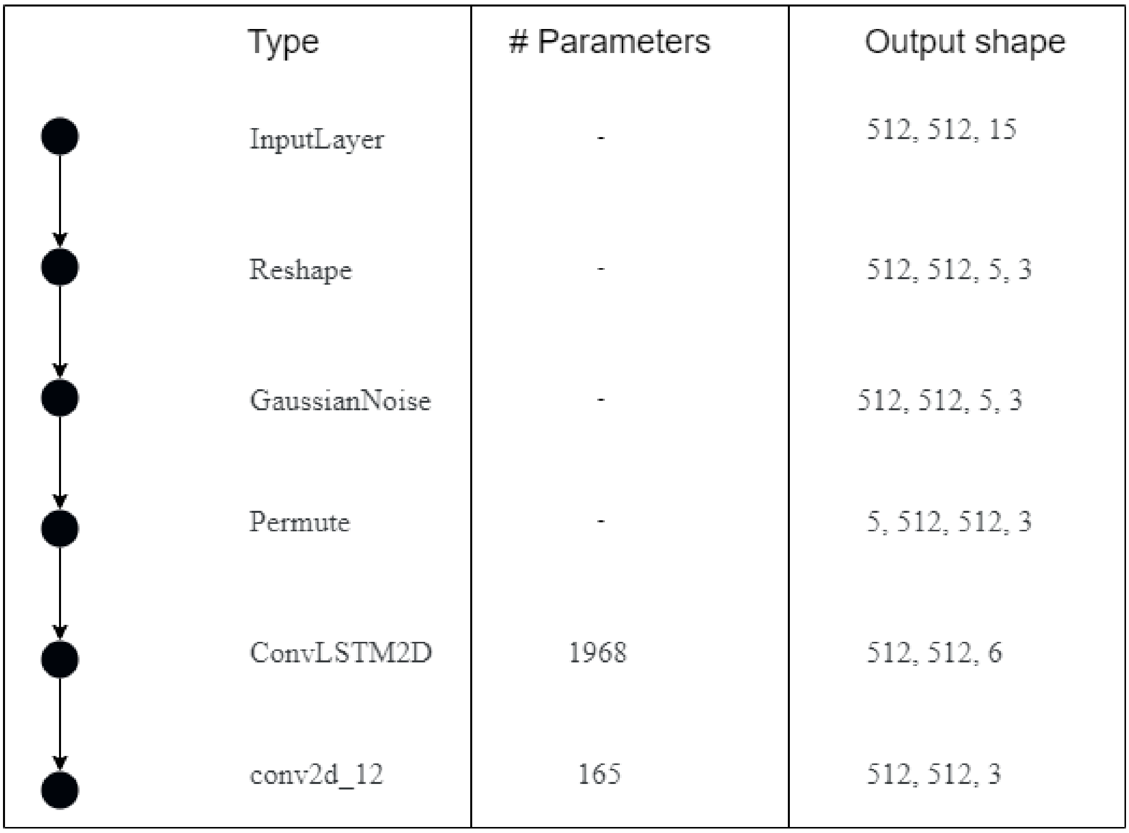
$$NParameters = (kW * kH * kD * nF) + nK \tag{C.1}$$

$$NParameters = 4 * oC * (kS * *2 * (iC + oC) + 1) \tag{C.2}$$

Symbol	Meaning	ConvLSTM	Conv3D
kW	Kernel Width	-	3
kH	Kernel Height	-	3
kD	Kernel Depth	-	3
nF	Number of Frames	-	6
nK	Number of Kernels	-	3
kS	Kernel Size	3	-
iC	Input Channels	3	-
oC	Output Channels	6	-

Table C.1: Symbols, meanings, and values of the parameters equations

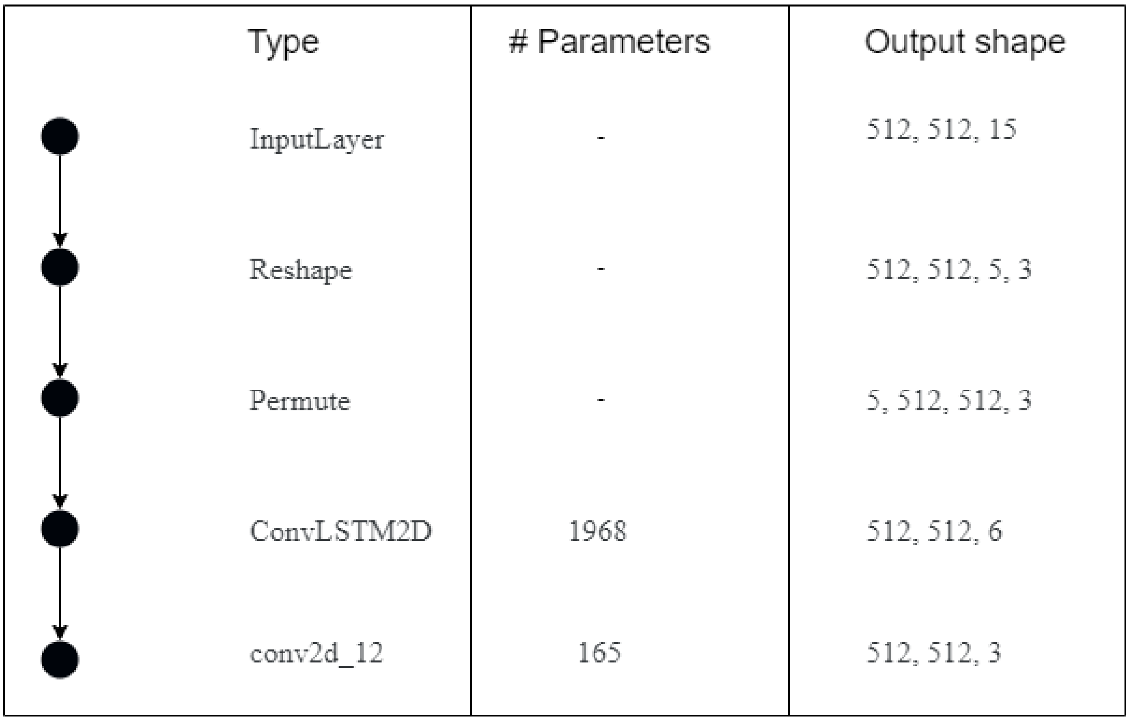
C.1 ConvLSTM with gaussian noise



Type	# Parameters	Output shape
InputLayer	-	512, 512, 15
Reshape	-	512, 512, 5, 3
GaussianNoise	-	512, 512, 5, 3
Permute	-	5, 512, 512, 3
ConvLSTM2D	1968	512, 512, 6
conv2d_12	165	512, 512, 3

FIGURE C.1: ConvLSTM with gaussian noise.

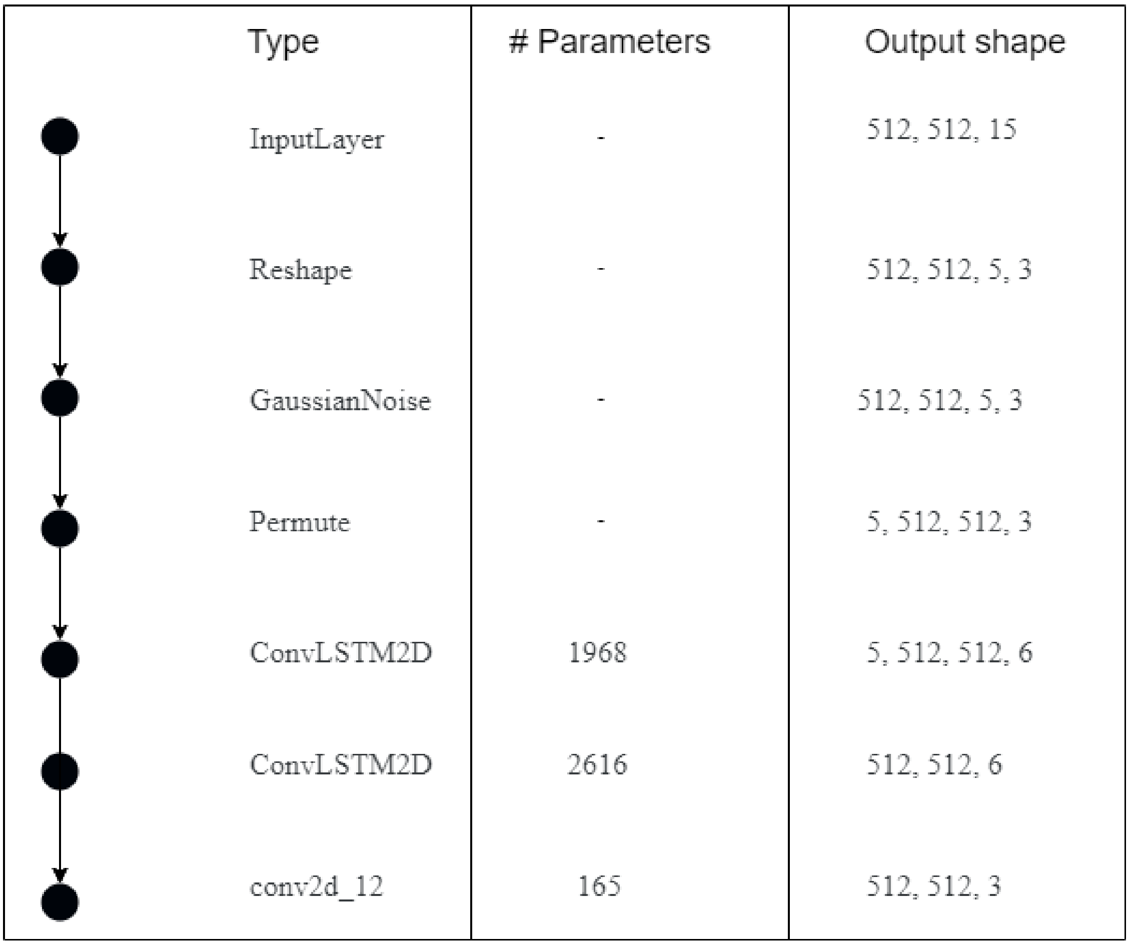
C.2 ConvLSTM withou gaussian noise



Type	# Parameters	Output shape
InputLayer	-	512, 512, 15
Reshape	-	512, 512, 5, 3
Permute	-	5, 512, 512, 3
ConvLSTM2D	1968	512, 512, 6
conv2d_12	165	512, 512, 3

Figure C.2: ConvLSTM withou gaussian noise.

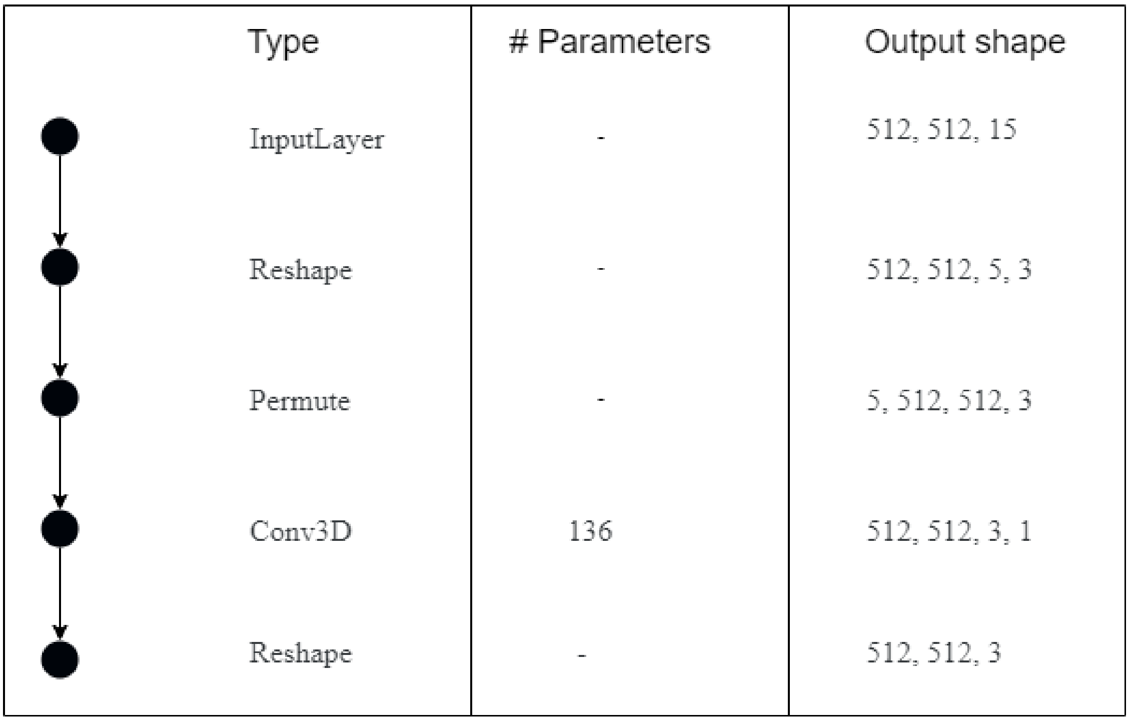
C.3 Double ConvLSTM without gaussian noise



Type	# Parameters	Output shape
InputLayer	-	512, 512, 15
Reshape	-	512, 512, 5, 3
GaussianNoise	-	512, 512, 5, 3
Permute	-	5, 512, 512, 3
ConvLSTM2D	1968	5, 512, 512, 6
ConvLSTM2D	2616	512, 512, 6
conv2d_12	165	512, 512, 3

FIGURE C.3: Double ConvLSTM without gaussian noise.

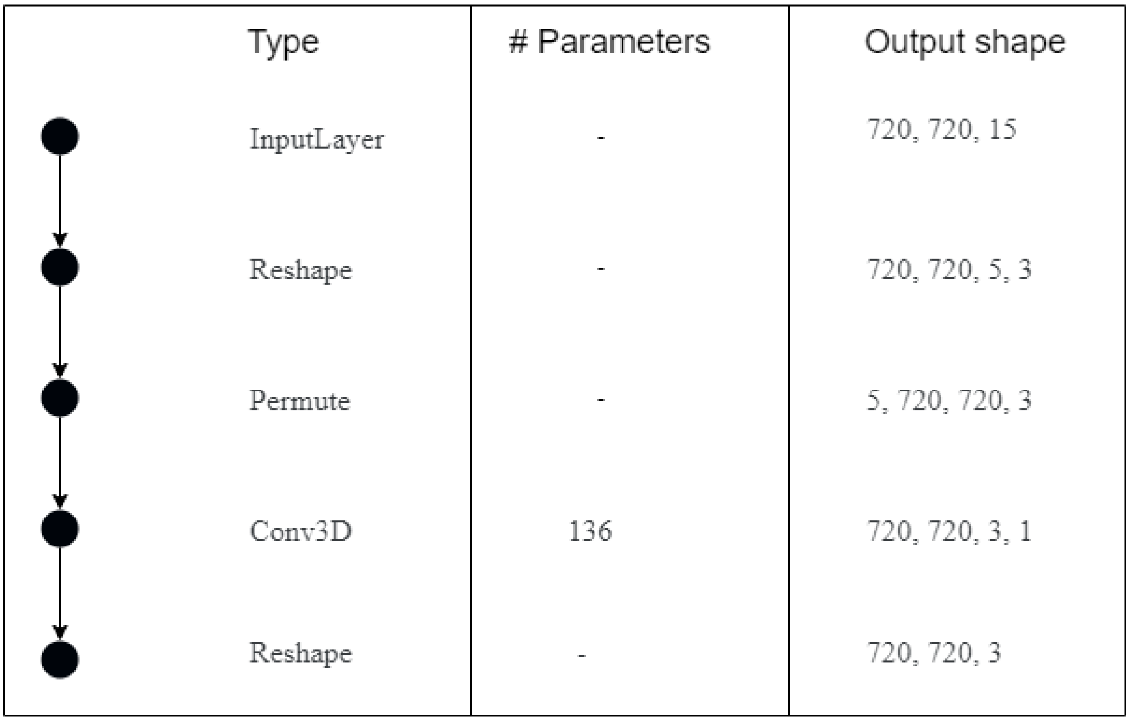
C.4 Conv3D 512 tile size



Type	# Parameters	Output shape
InputLayer	-	512, 512, 15
Reshape	-	512, 512, 5, 3
Permute	-	5, 512, 512, 3
Conv3D	136	512, 512, 3, 1
Reshape	-	512, 512, 3

Figure C.4: Conv3D 512 tile size.

C.5 Conv3D 720 tile size



Type	# Parameters	Output shape
InputLayer	-	720, 720, 15
Reshape	-	720, 720, 5, 3
Permute	-	5, 720, 720, 3
Conv3D	136	720, 720, 3, 1
Reshape	-	720, 720, 3

Figure C.5: Conv3D 720 tile size.

Bibliography

Abood, S. A., Lee, J. S. H., Burivalova, Z., Garcia-Ulloa, J., and Koh, L. P. (2015). Relative contributions of the logging, fiber, oil palm, and mining industries to forest loss in indonesia. *Conservation Letters*, 8(1):58–67.

Ahmadi, V. (2018). Deforestation prediction using neural networks and satellite imagery in a spatial information system. *arXiv preprint arXiv:1803.02489*.

Alencar, A., Nepstad, D., McGrath, D., Moutinho, P., Pacheco, P., Diaz, M. V., Soares Filho, B., et al. (2004). *Desmatamento na Amazônia: indo além da" emergência crônica"*. Ipam Belémˆ ePA PA.

Asner, G. P., Knapp, D. E., Broadbent, E. N., Oliveira, P. J., Keller, M., and Silva, J. N. (2005). Selective logging in the brazilian amazon. *science*, 310(5747):480–482.

Barber, C. P., Cochrane, M. A., Souza Jr, C. M., and Laurance, W. F. (2014). Roads, deforestation, and the mitigating effect of protected areas in the amazon. *Biological conservation*, 177:203–209.

Booth, B. and Mitchell, A. (2001). Getting started with arcgis.

Botvinick, M. M. and Plaut, D. C. (2006). Short-term memory for serial order: a recurrent neural network model. *Psychological review*, 113(2):201.

Brearley, F. Q., Adinugroho, W. C., Cámara-Leret, R., Krisnawati, H., Ledo, A., Qie, L., Smith, T. E., Aini, F., Garnier, F., Lestari, N. S., et al. (2019). Opportunities and challenges for an indonesian forest monitoring network. *Annals of Forest Science*, 76(2):54.

Brodersen, K. H., Ong, C. S., Stephan, K. E., and Buhmann, J. M. (2010). The balanced accuracy and its posterior distribution. In *2010 20th International Conference on Pattern Recognition*, pages 3121–3124. IEEE.

Buntine, W. L. and Weigend, A. S. (1991). Bayesian back-propagation. *Complex systems*, 5(6):603–643.

Burnaev, E., Erofeev, P., and Papanov, A. (2015). Influence of resampling on accuracy of imbalanced classification. In *Eighth International Conference on Machine Vision (ICMV 2015)*, volume 9875, page 987521. International Society for Optics and Photonics.

Caruana, R., Lou, Y., Gehrke, J., Koch, P., Sturm, M., and Elhadad, N. (2015). Intelligible models for healthcare: Predicting pneumonia risk and hospital 30-day readmission. In *Proceedings of the 21th ACM SIGKDD international conference on knowledge discovery and data mining*, pages 1721–1730.

Chongxuan, L., Xu, T., Zhu, J., and Zhang, B. (2017). Triple generative adversarial nets. In *Advances in neural information processing systems*, pages 4088–4098.

Das, K. and Behera, R. N. (2017). A survey on machine learning: concept, algorithms and applications. *International Journal of Innovative Research in Computer and Communication Engineering*, 5(2):1301–1309.

DeFries, R. and Chan, J. C.-W. (2000). Multiple criteria for evaluating machine learning algorithms for land cover classification from satellite data. *Remote Sensing of Environment*, 74(3):503–515.

Doshi-Velez, F. and Kim, B. (2017). Towards a rigorous science of interpretable machine learning. *arXiv preprint arXiv:1702.08608*.

Dunford, R., Su, Q., and Tamang, E. (2014). The pareto principle.

Gao, F., Huang, T., Wang, J., Sun, J., Hussain, A., and Yang, E. (2017). Dual-branch deep convolution neural network for polarimetric sar image classification. *Applied Sciences*, 7(5):447.

Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., and Bengio, Y. (2014). Generative adversarial nets. In *Advances in neural information processing systems*, pages 2672–2680.

Grieg-Gran, M. (2006). The cost of avoiding deforestation. *report prepared for Stern Review, International Institute for Environment and Development*.

Harry, S. (2016). The dawn of artificial inteligence.

Hassoun, M. H. et al. (1995). *Fundamentals of artificial neural networks*. MIT press.

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

Hinton, G. E., Osindero, S., and Teh, Y.-W. (2006). A fast learning algorithm for deep belief nets. *Neural computation*, 18(7):1527–1554.

Hochreiter, S. and Schmidhuber, J. (1997). Long short-term memory. *Neural computation*, 9(8):1735–1780.

Hong, X., Chen, S., and Harris, C. J. (2007). A kernel-based two-class classifier for imbalanced data sets. *IEEE Transactions on neural networks*, 18(1):28–41.

Huang, G., Liu, Z., Van Der Maaten, L., and Weinberger, K. Q. (2017). Densely connected convolutional networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4700–4708.

Ioffe, S. and Szegedy, C. (2015). Batch normalization: Accelerating deep network training by reducing internal covariate shift. *arXiv preprint arXiv:1502.03167*.

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

Krizhevsky, A., Sutskever, I., and Hinton, G. E. (2012). Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*, pages 1097–1105.

Kwaśniewska, A., Rumiński, J., and Rad, P. (2017). Deep features class activation map for thermal face detection and tracking. In *2017 10th International Conference on Human System Interactions (HSI)*, pages 41–47. IEEE.

Laurance, W. F., Cochrane, M. A., Bergen, S., Fearnside, P. M., Delamônica, P., Barber, C., D'angelo, S., and Fernandes, T. (2001). The future of the brazilian amazon. *Science*, 291(5503):438–439.

LeCun, Y., Bengio, Y., and Hinton, G. (2015). Deep learning. *nature*, 521(7553):436–444.

Lipton, Z. C. (2018). The mythos of model interpretability. *Queue*, 16(3):31–57.

Liu, T., Li, Y., Cao, Y., and Shen, Q. (2017). Change detection in multitemporal synthetic aperture radar images using dual-channel convolutional neural network. *Journal of Applied Remote Sensing*, 11(4):042615.

Lukoševičius, M. and Jaeger, H. (2009). Reservoir computing approaches to recurrent neural network training. *Computer Science Review*, 3(3):127–149.

Mello, N. G. R. d. and Artaxo, P. (2017). Evolução do plano de ação para prevenção e controle do desmatamento na amazônia legal. *Revista do Instituto de Estudos Brasileiros*, (66):108–129.

Meng, F., Huang, K., Li, H., and Wu, Q. (2019). Class activation map generation by representative class selection and multi-layer feature fusion. *arXiv preprint arXiv:1901.07683*.

Mirza, M. and Osindero, S. (2014). Conditional generative adversarial nets. *arXiv preprint arXiv:1411.1784*.

Mitchell, T. M. (2006). *The discipline of machine learning*, volume 9. Carnegie Mellon University, School of Computer Science, Machine Learning ....

Mittelman, R., Kuipers, B., Savarese, S., and Lee, H. (2014). Structured recurrent temporal restricted boltzmann machines. In *International Conference on Machine Learning*, pages 1647–1655.

Montana, D. J. and Davis, L. (1989). Training feedforward neural networks using genetic algorithms. In *IJCAI*, volume 89, pages 762–767.

Myers, N. (1991). Tropical forests: present status and future outlook. *Climatic change*, 19(1-2):3–32.

Nair, V. and Hinton, G. E. (2010). Rectified linear units improve restricted boltzmann machines. In *Proceedings of the 27th international conference on machine learning (ICML-10)*, pages 807–814.

Neal, R. M. (1992). Connectionist learning of belief networks. *Artificial intelligence*, 56(1):71–113.

Pendrill, F., Persson, U. M., Godar, J., Kastner, T., Moran, D., Schmidt, S., and Wood, R. (2019). Agricultural and forestry trade drives large share of tropical deforestation emissions. *Global environmental change*, 56:1–10.

Pope, P. E., Kolouri, S., Rostami, M., Martin, C. E., and Hoffmann, H. (2019). Explainability methods for graph convolutional neural networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 10772–10781.

Putz, F. E. and Redford, K. H. (2010). The importance of defining 'forest': Tropical forest degradation, deforestation, long-term phase shifts, and further transitions. *Biotropica*, 42(1):10–20.

Radford, A., Metz, L., and Chintala, S. (2015). Unsupervised representation learning with deep convolutional generative adversarial networks. *arXiv preprint arXiv:1511.06434*.

Reddy, C. S., Dutta, K., and Jha, C. (2013). Analysing the gross and net deforestation rates in india. *Current Science*, pages 1492–1500.

Reed, S. E., Akata, Z., Mohan, S., Tenka, S., Schiele, B., and Lee, H. (2016). Learning what and where to draw. In *Advances in neural information processing systems*, pages 217–225.

Ribeiro, M. T., Singh, S., and Guestrin, C. (2016). " why should i trust you?" explaining the predictions of any classifier. In *Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining*, pages 1135–1144.

Rogan, J., Franklin, J., Stow, D., Miller, J., Woodcock, C., and Roberts, D. (2008). Mapping land-cover modifications over large areas: A comparison of machine learning algorithms. *Remote Sensing of Environment*, 112(5):2272–2283.

Saito, T. and Rehmsmeier, M. (2015). The precision-recall plot is more informative than the roc plot when evaluating binary classifiers on imbalanced datasets. *PloS one*, 10(3):e0118432.

Sales, M., De Bruin, S., Herold, M., Kyriakidis, P., and Souza Jr, C. (2017). A spatiotemporal geostatistical hurdle model approach for short-term deforestation prediction. *Spatial Statistics*, 21:304–318.

Sandberg, I. W., Lo, J. T., Fancourt, C. L., Principe, J. C., Katagiri, S., and Haykin, S. (2001). *Nonlinear dynamical systems: feedforward neural network perspectives*, volume 21. John Wiley & Sons.

Sasaki, N. and Putz, F. E. (2009). Critical need for new definitions of "forest" and "forest degradation" in global climate change agreements. *Conservation Letters*, 2(5):226–232.

Schmidhuber, J. (1992). Learning factorial codes by predictability minimization. *Neural Computation*, 4(6):863–879.

Schmidhuber, J. (2015). Deep learning in neural networks: An overview. *Neural networks*, 61:85–117.

Scholz, I. (2006). 8. negotiating solutions for local sustainable development and the prevention of deforestation in the brazilian amazon. *Partnerships in Sustainable Forest Resource Management: Learning from Latin America*, pages 279–300.

Shi, X. and Yeung, D.-Y. (2018). Machine learning for spatiotemporal sequence forecasting: A survey. *arXiv preprint arXiv:1808.06865*.

Smolensky, P. (1986). Information processing in dynamical systems: Foundations of harmony theory. Technical report, Colorado Univ at Boulder Dept of Computer Science.

Stone, T. A. and Lefebvre, P. (1998). Using multi-temporal satellite data to evaluate selective logging in para, brazil. *International Journal of Remote Sensing*, 19(13):2517–2526.

Sutskever, I. and Hinton, G. (2007). Learning multilevel distributed representations for high-dimensional sequences. In *Artificial intelligence and statistics*, pages 548–555.

Sutskever, I., Hinton, G. E., and Taylor, G. W. (2009). The recurrent temporal restricted boltzmann machine. In *Advances in neural information processing systems*, pages 1601–1608.

Thangarajah, A. (2019). Computing total number of parameters in conv3d and convlstm2d.

Torahi, A. A. (2013). Forest mapping and change analysis, using satellite imagery in zagros mountain, iran. *Lebanese Science Journal*, 14(2):63.

van de Weerd, I. and Brinkkemper, S. (2009). Meta-modeling for situational analysis and design methods. In *Handbook of research on modern systems analysis and design technologies and applications*, pages 35–54. IGI Global.

Vondrick, C., Pirsiavash, H., and Torralba, A. (2016). Generating videos with scene dynamics. In *Advances in neural information processing systems*, pages 613–621.

Wang, T.-C., Liu, M.-Y., Zhu, J.-Y., Tao, A., Kautz, J., and Catanzaro, B. (2018). High-resolution image synthesis and semantic manipulation with conditional gans. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 8798–8807.

Wibowo, D. H. and Byron, R. N. (1999). Deforestation mechanisms: A survey. *International Journal of Social Economics*.

Wieringa, R. J. (2014). *Design science methodology for information systems and software engineering*. Springer.

Xingjian, S., Chen, Z., Wang, H., Yeung, D.-Y., Wong, W.-K., and Woo, W.-c. (2015). Convolutional lstm network: A machine learning approach for precipitation nowcasting. In *Advances in neural information processing systems*, pages 802–810.

Xu, Z., Du, J., Wang, J., Jiang, C., and Ren, Y. (2019). Satellite image prediction relying on gan and lstm neural networks. In *ICC 2019-2019 IEEE International Conference on Communications (ICC)*, pages 1–6. IEEE.

Yang, C., Rangarajan, A., and Ranka, S. (2018). Visual explanations from deep 3d convolutional neural networks for alzheimer's disease classification. In *AMIA Annual Symposium Proceedings*, volume 2018, page 1571. American Medical Informatics Association.

Zhang, Q., Nian Wu, Y., and Zhu, S.-C. (2018). Interpretable convolutional neural networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 8827–8836.