Pontifícia Universidade Católica
do Rio de Janeiro

Jorge Andres Chamorro Martinez

Many-to-Many Fully Convolutional Recurrent Networks for Multitemporal Crop Recognition Using SAR Image Sequences

Dissertação de Mestrado

Thesis presented to the Programa de Pós–graduação em Engenharia Elétrica of PUC-Rio in partial fulfillment of the requirements for the degree of Mestre em Engenharia Elétrica.

Advisor: Prof. Raul Queiroz Feitosa

Rio de Janeiro
August 2019

Pontifícia Universidade Católica
do Rio de Janeiro

Jorge Andres Chamorro Martinez

Many-to-Many Fully Convolutional Recurrent Networks for Multitemporal Crop Recognition Using SAR Image Sequences

Thesis presented to the Programa de Pós–graduação em Engenharia Elétrica of PUC-Rio in partial fulfillment of the requirements for the degree of Mestre em Engenharia Elétrica. Approved by the Examination Committee.

Prof. Raul Queiroz Feitosa
Advisor
Departamento de Engenharia Elétrica – PUC-Rio

Prof. Wouter Caarls
Departamento de Engenharia Elétrica – PUC-Rio

Prof. Ieda Del'Arco Sanches
INPE

Dra. Cristina Maria Bentz
CENPES – PETROBRAS

Dr. Cleber Gonzales de Oliveira
Visiona Tecnologia Espacial – Visiona

Rio de Janeiro, August the 27th, 2019

Jorge Andres Chamorro Martinez

The author received his bachelor's degree in Electronic Engineering at the University of Nariño in 2015. He worked for 2 years using machine learning, computer vision and embedded systems for applications in the areas of agriculture, security, health and sports. Since then, he has worked in the fields of machine learning, computer vision and remote sensing during his Master's degree studies at the Pontifical Catholic University of Rio de Janeiro (PUC-Rio).

Bibliographic data

Chamorro Martinez, Jorge Andres

Many-to-Many Fully Convolutional Recurrent Networks for Multitemporal Crop Recognition Using SAR Image Sequences / Jorge Andres Chamorro Martinez; advisor: Raul Queiroz Feitosa. – Rio de janeiro: PUC-Rio , Departamento de Engenharia Elétrica, 2019.

v., 67 f: il. color. ; 30 cm

Dissertação (mestrado) - Pontifícia Universidade Católica do Rio de Janeiro, Departamento de Engenharia Elétrica.

Inclui bibliografia

1. Engenharia Elétrica – Teses. 2. Redes Totalmente Convolucionais;. 3. Redes Recorrentes;. 4. Reconhecimento de Culturas;. 5. Sensoriamento Remoto. I. Feitosa, Raul Queiroz. II. Pontifícia Universidade Católica do Rio de Janeiro. Departamento de Engenharia Elétrica. III. Título.

CDD: 621.3

Acknowledgments

I thank my advisor Prof. Raul Queiroz Feitosa for all the lessons learned in the areas of engineering, teaching and critical thinking. For the encouragement and trust he has given me to pursue interesting and relevant projects under his supervision.

I also thank my family for their unconditional love and support, and for making me feel as if I was at home every day through the use of technology.

I thank my laboratory colleagues at the Computer Vision Lab (LVC), for all the help and the happy memories.

Finally I acknowledge the valuable financial support of CNPq.

Abstract

Chamorro Martinez, Jorge Andres; Feitosa, Raul Queiroz (Advisor). **Many-to-Many Fully Convolutional Recurrent Networks for Multitemporal Crop Recognition Using SAR Image Sequences**. Rio de Janeiro, 2019. 67p. Dissertação de mestrado – Departamento de Engenharia Elétrica, Pontifícia Universidade Católica do Rio de Janeiro.

This work proposes and evaluates deep learning architectures for multi-date agricultural crop recognition from remote sensing image sequences. These architectures combine the spatial modelling capabilities of fully convolutional networks and the sequential modelling capabilities of recurrent networks into end-to-end architectures so-called fully convolutional recurrent networks, configured to predict crop type at multiple dates from a multitemporal image sequence. Their performance is assessed over two publicly available datasets. Both datasets present highly spatio-temporal dynamics due to their tropical/sub-tropical climate and local agricultural practices such as crop rotation. The experiments indicated that the proposed architectures outperformed state of the art methods based on recurrent networks in terms of *Overall Accuracy* (OA) and per-class average F1 score.

Keywords

Fully Convolutional Networks; Recurrent Networks; Crop Recognition; Remote Sensing

Resumo

Chamorro Martinez, Jorge Andres; Feitosa, Raul Queiroz. **Reconhecimento de culturas agrícolas utilizando redes recorrentes a partir de sequências de imagens SAR**. Rio de Janeiro, 2019. 67p. Dissertação de Mestrado – Departamento de Engenharia Elétrica, Pontifícia Universidade Católica do Rio de Janeiro.

Este trabalho propõe e avalia arquiteturas profundas para o reconhecimento de culturas agrícolas a partir de seqüências de imagens multitemporais de sensoriamento remoto. Essas arquiteturas combinam a capacidade de modelar contexto espacial prórpia de redes totalmente convolucionais com a capacidade de modelr o contexto temporal de redes recorrentes para a previsão prever culturas agrícolas em cada data de uma seqüência de imagens multitemporais. O desempenho destes métodos é avaliado em dois conjuntos de dados públicos. Ambas as áreas apresentam alta dinâmica espaço-temporal devido ao clima tropical/subtropical e a práticas agrícolas locais, como a rotação de culturas. Nos experimentos verificou-se que as arquiteturas propostas superaram os métodos recentes baseados em redes recorrentes em termos de *Overall Accuracy* (OA) e F1-*score* médio por classe.

Palavras-chave

Redes Totalmente Convolucionais; Redes Recorrentes; Reconhecimento de Culturas; Sensoriamento Remoto

Table of contents

List of figures

List of tables

List of Abreviations

ASPP	Atrous Spatial Pyramid Pooling
BAtrousConvLSTM	Bidirectional Atrous Convolutional Long Short-Term Memory
BConvLSTM	Bidirectional Convolutional Long Short-Term Memory
BDenseConvLSTM	Bidirectional Dense Convolutional Long Short-Term Memory
BUnetConvLSTM	Bidirectional U-Net Convolutional Long Short-Term Memory
BN	Batch Normalization
ConvLSTM	Convolutional Long-Short Term Memory
CNN	Convolutional Neural Networks
CRF	Conditional Random Fields
DB	Dense Block
DL	Deep Learning
FCN	Fully Convolutional Networks
GAP	Global Average Pooling
GLCM	Gray-Level Co-occurrence Matrix
GRU	Gated Recurrent Unit
GPU	Graphics Processing Unit
H	Horizontal
HH	Horizontal-Horizontal
HMM	Hidden Markov Models
HV	Horizontal-Vertical
KNN	K-Nearest Neighbor
LEM	Luiz Eduardo Magalhães
LSTM	Long Short-Term Memory
MLP	Multi-Layer Perceptron
MRF	Markov Random Fields

NCC	Non-Commercial Crops
NDVI	Normalized Difference Vegetation Index
NN	Neural Network
OA	Overall Accuracy
OBIA	Object-Based Image Analysis
RADAR	Radio Detection And Ranging
ReLU	Rectified Linear Unit
RF	Random Forest
RNN	Recurrent Neural Networks
RS	Remote Sensing
SAR	Synthetic Aperture Radar
SVM	Support Vector Machine
TD	Transition Down
TU	Transition Up
UConvLSTM	Unidirectional Convolutional Long Short-Term Memory
UV	Ultraviolet
V	Vertical
VH	Vertical-Horizontal
VV	Vertical-Vertical

1 INTRODUCTION

1.1 Motivation

The projections of world population for the next decades demand more efficient, comprehensive and precise agriculture. According to the United Nations reports, the world population is expected to reach 8.6 billion by 2030, 9.8 billion by 2050 and 11.2 billion by 2100 [8]. It is therefore necessary to promote policies to increase global agricultural production to ensure food supply with minimal environmental impact. In this context, crop monitoring is very important to develop commercial plans, regulate internal stocks and perform customized management decisions [9]. Crop recognition task is important because it is needed to obtain other relevant information such as prediction of crop yield. Multitemporal remote sensing (RS) imagery has increasingly been applied for this task as a cost-effective way for gathering timely, detailed and reliable information over large areas [10].

Crop recognition from RS data is particularly challenging in tropical regions, because the favorable climate associated with the use of modern technologies makes agriculture highly dynamic [11].

In recent years, deep learning models have made breakthroughs in several fields such as speech recognition and computer vision [12]. In remote sensing, these models have also been successfully tested in diverse applications [13]. Such models can be roughly grouped in two main categories: Convolutional Neural Networks (CNN) for understanding spatial context, and Recurrent Neural Networks (RNN), mostly to model data sequences.

In [14], a type of CNN called Fully Convolutional Network (FCN) was used for crop recognition having as input the stack of a multi-temporal sequence. Although a good performance is reported, the method requires the training of a particular model for each date. Thus, this solution can become computationally expensive in many practical applications.

RNNs can be configured to allow sequential inputs and to produce a single outcome that represents the semantic of the whole input sequence. Such "many-to-one" configurations have been used for crop-recognition in temperate regions, where a single crop occurs in each field over the whole season.

In [15] two different RNN models, Long short-term memory (LSTM) and Gated Recurrent Unit (GRU), were applied for crop classification upon multi-temporal Sentinel-1 data. In [16], a CNN was proposed to provide the input to a RNN for the many-to-one crop recognition task.

In [17], the internal fully connected LSTM layers were replaced by convolutional layers. This type of recurrent convolutional network (ConvLSTM) is able to jointly model the spatial and temporal context from multi-temporal sequences of images. This kind of RNN was used for precipitation forecasting. Later, in [18], this ConvLSTM network was applied to the multi-temporal land cover classification problem in a many-to-one configuration. Furthermore, the same work used a bidirectional variant of ConvLSTM to eliminate bias toward the later sequence elements. All aforementioned proposals follow the many-to-one approach.

In areas with complex crop dynamics, such as in tropical regions, multiple crops may come about in a field during the season. Thus, the single crop per season assumption does not hold in those regions. Therefore, networks capable of performing crop recognition at multiple dates are required.

Our work hypothesis is that many-to-many RNN configurations can be applied for multidate crop recognition, to accurately identify crop classes in tropical regions at each date represented in a multitemporal sequence. Specifically, we introduce a novel many-to-many configuration of a bidirectional ConvLSTM for multidate crop recognition from multitemporal RS data [18].

A limitation of the ConvLSTM approach from [18] is that it computes convolutions at a single spatial scale. In contrast, modern FCN architectures are designed to extract features at multiple spatial scales by successively reducing the input image resolution or increasing the convolution kernel size. This master thesis proposes hybrid architectures combining the FCN multi-scale feature extraction capabilities with the ConvLSTM spatio-temporal modeling properties. For these hybrid architectures, some of the most relevant FCN approaches were considered: U-Net, Dense FCN and *Atrous Spatial Pyramid Pooling* (ASPP).

The first proposed network uses a U-Net encoder to provide inputs at a lower spatial resolution to a bidirectional ConvLSTM. After processing the input provided by the encoder, the ConvLSTM delivers the output, which is then applied to a decoder that generates the outcome, a pixel-wise label image, at the original spatial resolution. The second architecture is a variant of the first one, which comprises additional internal connections. The third architecture applies an ASPP feature layer to the inputs extracting multi-scale spatial features, which are fed to a bidirectional ConvLSTM for spatio-temporal feature extraction.

In addition, two convolutional many-to-one RNNs, introduced in earlier works [18], were adapted to the many-to-many task and compared with the proposed hybrid architectures. The experiments were carried out upon datasets of two tropical regions charaterized by complex spatio-temporal dynamics and crop rotation practices.

To the best of our knowledge, this is the first work that addresses many-to-many convolutional recurrent networks as unique, end-to-end architectures, for pixel-wise crop recognition of entire image sequences.

1.2 Objectives

1.2.1 General Objective

The general objective of this work is to propose a model for crop recognition in tropical regions based on a hybrid deep learning approach leveraging fully convolutional and recurrent networks for sequences of remote sensing images.

1.2.2 Specific Objectives

The specific objectives of this work are the following:

1. Design a unique, end-to-end deep learning network capable of producing pixel-wise classifications for entire sequences of remote sensing images for crop recognition applications.

2. Consider spatial and temporal context in the designed architecture using concepts from recurrent and fully convolutional networks.

3. Test the designed network performance on study areas with highly dynamic spatio temporal crop dynamics.

1.3 Contributions

The main contributions of this work are the following:

1. A novel recurrent network architecture that combines bidirectional LSTM and FCN for multidate crop recognition

2. A performance assessment of some of the latest FCN architectures for the proposed recurrent network.

3. An extension of convolutional LSTMs originally designed for single crop per season applications to multidate crop recognition

4. An experimental analysis of the aforementioned network designs on datasets that represent highly dynamic agriculture typical of tropical regions.

1.4 Organization of the remaining parts of this thesis

Chapter 2 describes some of the most relevant approaches for multi-temporal crop recognition using sequences of remote sensing images with a focus on tropical environments. The main categories of these approaches are OBIA, probabilistic graphical models, convolutional networks and recurrent networks.

Chapter 3 describes the basic concepts and theory required to understand the methods proposed in this work, including fully convolutional networks and recurrent networks.

Chapter 4 presents the methods for many-to-many multi-temporal recognition, including the proposed hybrid fully convolutional recurrent architectures.

Chapter 5 details the experiments and their protocol including the study areas and the network hyper-parameter configurations. Furthermore, the results of those experiments are presented and discussed.

Chapter 6 summarizes the insights obtained from the experimental results and outlines future lines of work that could be further researched.

2 RELATED WORKS

This chapter presents an overview of different works applied to crop recognition from multi-temporal satellite image sequences. First, classical remote sensing methods like pixe-wise classification from vegetation indexes are briefly explained. Then works related with deep learning are presented with a focus on convolutional and recurrent networks.

Traditional remote sensing image analysis approaches use the spectral information from each individual pixel location as the unit of analysis. In this case, a supervised classifier is trained using pixels as individual training samples. Different types of classifiers have been successfuly used such as Random Forest (RF), Support Vector Machines (SVM) and K-Nearest Neighbors (KNN) [19–21]. Rather than using the original pixel information, some works perform manual feature extraction upon each pixel location to obtain more discriminative representations. These features could be vegetation indexes such as NDVI [22–24]. However, these approaches ignore any spatial relationship among neighboring pixels. Some works have considered this spatial context by extracting texture features such as the ones based on Gray Level Co-Occurence Matrix [22]. In other works, this context has been successfuly extracted from SAR data using polarimetric target decomposition [25]. Although this results in a performance improvement, their discriminative level is limited.

With the increase of spatial resolution in remote sensing images, object based image analysis (OBIA) has been proposed as an alternative to per-pixel analysis, aimed at defining objects that are made up of groups of pixels with similar spectral characteristics [26]. In general, this method has proved to improve results compared to pixel based approaches in images with high spatial resolution (minor to 10m). In images with larger spatial resolution (between 10m and 100m), some works have found OBIA to improve the results while others didn't find any improvement [23, 27, 28]. These approaches aren't specifically designed to model the temporal dynamics from agricultural crops.

Probabilistic graphical models, such as Markov Random Fields (MRF) and Conditional Random Fields (CRF) have successfuly been applied to multi-temporal crop mapping. These models are able to capture the crops spatio-temporal dynamics [29, 30]. However, these methods require an additional feature extraction step, which usually relies on hand-crafted features.

Recently, deep learning techniques have achieved state of the art performance in multiple applications including remote sensing image analysis [31]. Particularly, convolutional neural networks (CNN) are able to automatically learn feature representations which encode spectral and spatial information from the original images. These networks have been used for crop mapping, improving the results with respect to aforementioned approaches [32, 33]. In [32], a CNN learns high-level features from hyperspectral images that feed a Multi-Layer Perceptron (MLP) which assign crop classes to the image sites. In [33], a 2-d CNN applied in the spatial domain was compared with a 1-d CNN applied in the temporal domain. Both approaches achieved higher performance compared to the classical RF and MLP models. In these cases, each pixel was represented by the image patch centered on it A classifier was applied in a sliding-window manner over all the image delivering a pixel-wise classification outcome. Although this method effectively assigns a semantic class label to every pixel of the original image, it is computationally expensive because it involves a lot of redundant operations.

CNNs were originally designed for image classification. An extension of CNNs called Fully Convolutional Network (FCN) predicts class labels for individual pixels of an input image, making it efficient for semantic segmentation. Starting with the work of Long and co-authors [34], several FCNs architectures have been proposed and adapted to remote sensing applications [14, 35, 36]. In [35], a type of FCN called U-Net was used for multi-temporal crop mapping from Sentinel-1 products in temperate regions. Similarly, in [14] a modification of the U-Net architecture called Dense FCN was used for multi-temporal crop recognition in a sub-tropical environment. Although these networks achieved a high accuracy, they require to train a separate neural network for each date represented in the sequence. This implies in high computational complexity that increases with the sequence length.

Recurrent neural networks (RNN) were specifically conceived to process sequential information such as time series data. However, the original RNN design (also known as vanilla RNN) is only capable to exploit representations of recent input events. In contrast, two RNN variants: Long Short Term Memory (LSTM) and Gated Recurrent Unit (GRU) can preserve representations of much earlier events [37, 38]. These networks have been successfuly applied for agricultural crop mapping [15, 39, 40]. In [39], the vanilla RNN, LSTM and GRU networks were used for multi-temporal crop recognition. LSTM and GRU architectures presented a performance gain in relation to the original RNN design. Similarly, in [15] LSTM and GRU outperformed KNN, RF and

SVM approaches in multi-temporal crop classification upon Sentinel-1 data.

A weakness of all RNN variants in their original design is that they only consider temporal context but disregard spatial context. Contrarily, CNN can easily take spatial context into account but were not conceived to consider temporal dependencies. Hybrid approaches have been proposed to exploit the strengths of both concepts. In [16] a CNN was proposed to provide feature representations as inputs to a RNN using Sentinel-1 images. This network classifies an entire input image patch, and assigns its value to the patch central pixel. At test time, the network is applied to the image patches surrounding each test pixel location. As with CNNs, this approach is computationally expensive because too many redundant operations are needed. The authors of [17] proposed a LSTM variant replacing all its internal operations by convolutions (ConvLSTM). This type of network is inherently able to model the spatio-temporal dependencies of a multi-temporal sequence of images. In [18], the ConvLSTM was used for multi-temporal crop recognition in a temperate region.

RNNs can be configured to allow sequential inputs and to produce a single outcome that represents the semantic of the whole input sequence. Such "many-to-one" configurations have been used for crop-recognition in temperate regions, where a single crop occurs in each field over the whole season. All aforementioned RNN proposals follow the "many-to-one" approach. Such proposals are however inappropriate to model complex crop dynamics typical of tropical and sub-tropical environments, where different crop types may come about during a season. The present master thesis aims at filling this gap by proposing hybrid deep network architectures that combine RNN and FCN designs for crop mapping in a date-by-date basis, a so called "many-to-many" design.

3
FUNDAMENTALS

This chapter aims to provide a concise description of the concepts needed to understand the approaches proposed in this master thesis for multi-temporal crop recognition. First, a brief introduction to the SAR imaging sensor is presented, as well as its pros and cons in relation to optical sensors. Then the basic building blocks of deep learning architectures are presented, followed by a description of the most relevant Fully Convolutional Network (FCN) architectures for the problem of multi-temporal crop recognition. Finally, the fundamentals of recurrent neural networks (RNNs) and their variants Long-Short Term Memory (LSTM) and Convolutional Long-Short Term Memory (ConvLSTM) are described.

3.1
Synthetic Aperture Radar (SAR)

Remote sensing sensors are diverse and designed to capture different wavelength ranges from the electromagnetic radiation spectrum. Specifically, optical sensors perceive the earth's electromagnetic radiation close to the optical spectrum including ultra-violet, thermal and infrared ranges. However, such measurements are partially affected by earth's atmospheric conditions such as cloud coverage and weather conditions. In contrast, SAR sensors operate in the microwave electromagnetic range, which makes them capable of penetrating the atmosphere under most conditions [41]. This is illustrated in Figure 1, where atmospheric transmittance is presented as a function of wavelength (Transmittance is the effectiveness of a material in transmitting radiant energy [42]). There is very few atmospheric absorption in the microwave range, where SAR sensors operate.

While the earth does emit its own level of microwave radiation, it is often too small to be measured for most remote sensing purposes. SAR sensors work by sending microwave electromagnetic pulses to the earth and perceiving echoes of the energy that is scattered back to the sensing platform (Figure 2). Then the information of nearby back-scattered pulses is combined into image-like data [41]. The signal wavelength strongly influences the resulting image. For agriculture applications, the C-band (central frequency 5.4 GHz) has been used in multiple works [43].

Typically, radar signals are transmitted in a plane of polarization that is either parallel to the antenna axis (horizontal polarization, H) or perpendicular to that axis (vertical polarization, V). Likewise, the radar antenna may be set to receive only signals with a specified polarization. This results in four typical polarization combinations (HH, VV, HV, and VH), where the first letter indicates the transmitted polarization and the second indicates the received polarization. Because various objects modify the polarization of the energy they reflect to varying degrees, the mode of signal polarization influences how the objects look on the resulting imagery. Thus, each polarization combination may bring unique representations of the studied area [1]. A sample SAR image with multiple polarizations for an agricultural area is presented in Figure 3.

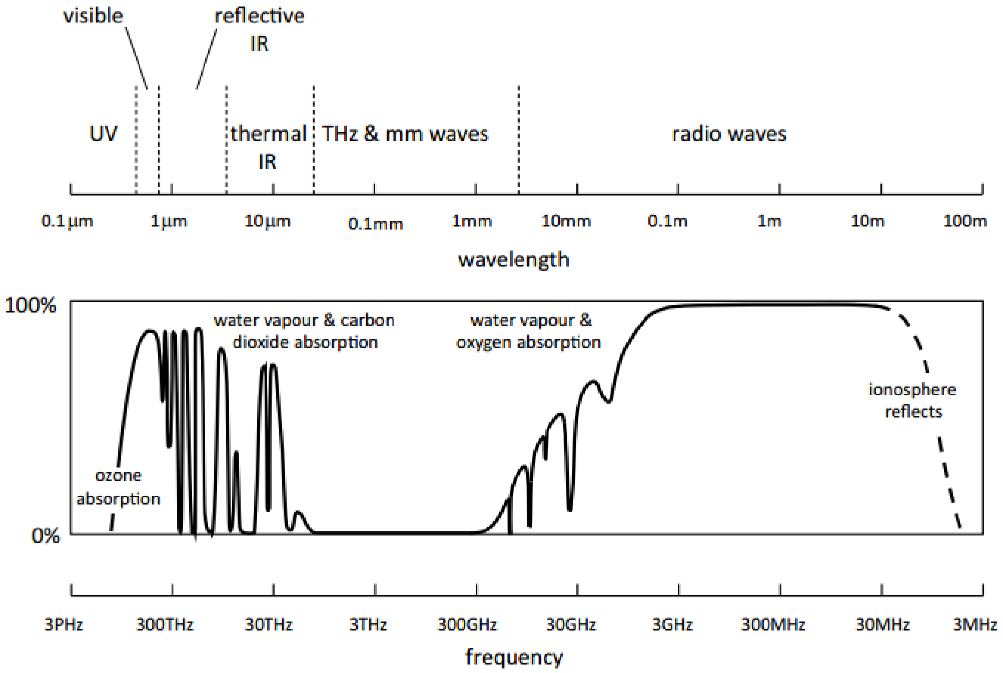

Figure 1: The electromagnetic spectrum and the earth's atmosphere transmittance. Transmittance is close to 100% for microwave (radio) waves, which are relatively unaffected by earth's atmospheric conditions [1].

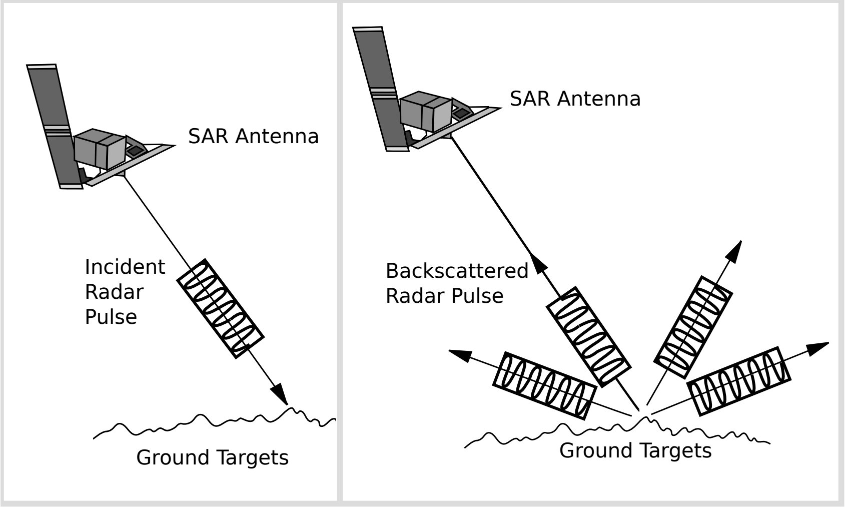

Figure 2: Radar working principle. First an electromagnetic signal in the microwave range is transmitted from the platform. Then the energy scattered back to the platform at microwave wavelengths is recorded (Adapted from [2]).

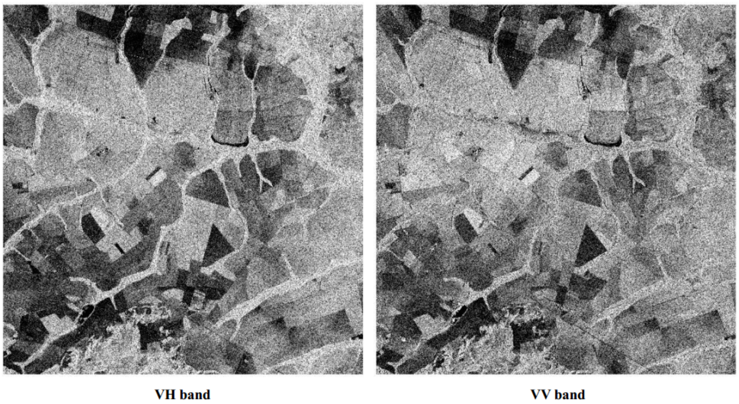

Figure 3: Sample SAR image from an agricultural area (VH and VV polarizations) [3].

3.2
Convolutional Neural Networks (CNNs)

A regular neural network consists of a series of fully connected layers, whereby each neuron of a layer is connected to all the neurons from the previous layer. This means, that every output unit interacts with every input unit. Depending on data being analyzed and on the number of layers this type of network may involve an excessive number of parameters, whose estimation through training may be very computationally expensive. Convolutional Neural Networks (CNNs) are a type of neural network specialized for processing data with grid-like structures such as time series (1D grid with regular time intervals) or images (2D grid of pixels). They replace the fully connected layers by convolutional layers, which do not require that each neuron in one layer is connected to all the neurons of the preceding layer. This implies in a comparatively much smaller amount of training parameters.

A typical CNN architecture is presented in Figure 4. It contains convolutional layers, followed by pooling operations and finally a fully connected layer with a softmax activation function. These operations are explained next.

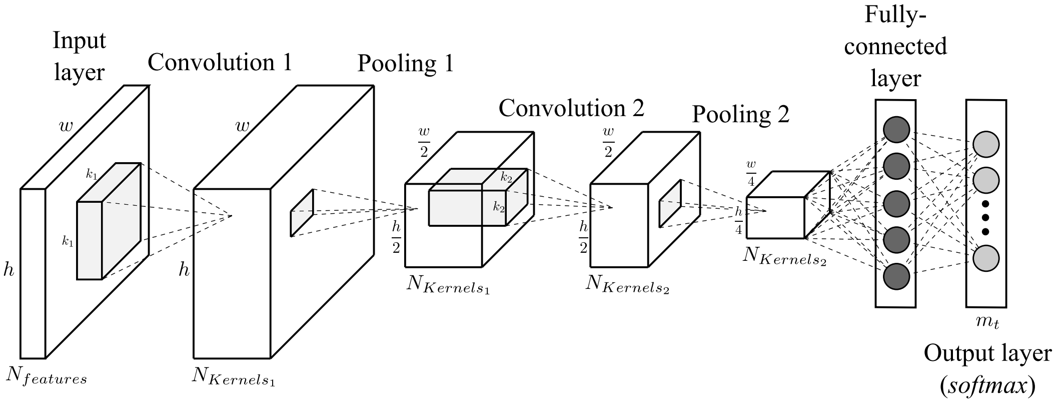

Figure 4: A CNN basic architecture with two convolutional layers [4].

Convolutional layer

Input to a convolutional layer is of dimensions $m \times n \times N_{input\_features}$, where m and n are the input spatial dimensions and $N_{input\_features}$ is the number of input feature maps. A convolution is applied to the input with a defined number of kernels of size $k \times k \times 1$, where the number of kernels corresponds to the amount of output feature maps and k is the kernel length. A convolution operation consists of sliding the kernel over the input image. At every location, an element-wise matrix multiplication with the kernel elements is performed which results are added up to the output feature representation. Generally, an activation function is applied to the resulting feature map.

Activation functions

Activation functions are non-linear mathematical operations typically applied at the output of internal layers in a multi-layer neural network. These functions introduce non-linearities which give the network capabilities to accurately approximate arbitrarily complex functions [44]. Some of the most common activation functions are *Sigmoid*, *tanh*, *ReLU* and *Leaky ReLU*.

The mathematical definitions and plot figures for each of these functions are presented in Figure 5.

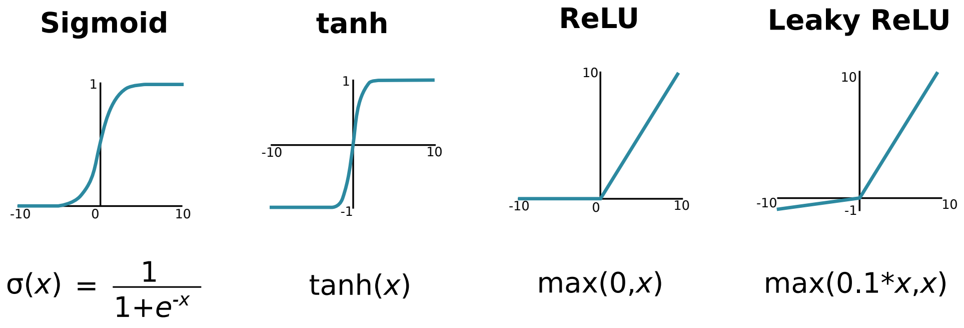

Figure 5: Mathematical definition and signal waveform for some of the most common activation functions.

Pooling layer

Input to a pooling layer is a tensor comprising the feature maps produced by the convolution operations carried out in the prior layer. This operation is typically used after a convolutional layer to reduce the feature map's spatial size and consequently minimize the amount of training parameters and the computational complexity. The most common pooling operation is a 2 × 2 max. pooling. It replaces each 2 × 2 tile, by the maximum value within that tile. An alternative operation is 2 × 2 average pooling, which replaces each 2 × 2 tile with its average value. These operations are applied separately to each feature map, modifying the spatial dimensions while the number of feature maps remains the same.

Batch Normalization

A problem with multi-layer neural networks is that the distribution of the input at each layer varies during training because the weights in the preceding layers are repeatedly being adjusted. This makes training difficult, particularly for layers whose activation functions saturate for some input values. Batch Normalization (BN) [45] aims to mitigate this problem by normalizing the values at layer inputs. The normalization parameters are learned to force each training batch to have zero mean and unit variance. This operation reduces the dependency of training on the parameter initialization and improves convergence.

Fully-connected layer

Fully connected layers connect its neurons to every output from the previous layer, as in the traditional Multi-Layer Perceptron (MLP). They are computed by multiplying the inputs with a weight matrix and adding a bias offset vector.

Dropout

Dropout is a method to reduce over-fitting [46]. The idea is to randomly deactivate neurons with their corresponding connections during training. In training, these deactivations result in multiple smaller versions of the original layer. At test time, all the units are used. Dropout can be interpreted as a way to emulate an ensemble of smaller networks with shared parameters.

Softmax function

The softmax operation is applied at the end of the network as a post-processing step to obtain a normalized vector of class probabilities at the output. In a CNN, the last layer is usually a fully connected layer with the amount of neurons equal to the number of classes. The softmax normalization ensures that the sum of these neurons is 1 and each of them is positive. In other words, the output of this layer can be seen as a probability distribution [47, 48]. The mathematical definition of this function is:

$$a_j = \frac{e^{z_j}}{\sum_{k=1}^{K} e^{z_k}} \quad for \;\; j = 1, 2, 3..., K \tag{3-1}$$

Where a_j is the activation result, z_j is the value of the j-th element in the vector to which the function is applied and K is the vector length, which should be equal to the amount of classes.

3.3 Fully Convolutional Networks (FCN)

A FCN is an extension of CNN designed to assign a semantic label to all pixels of the input image. Typical CNNs were designed for image classification tasks, and they don't produce pixel-wise classification outputs. A Fully Convolutional Network is an extension of CNNs designed to produce classifications for every pixel in the input image. An FCN replaces the fully connected at the end layer of a typical CNN with convolutional layers, which combined with upsampling operations provide the final output, a pixel classification map with the same spatial dimension as the input image.

A number of FCN architectures have been proposed in the last few year. In the following we describe succinctly the network architectures our proposals built upon.

3.3.1 Fully Convolutional U-Net

The U-Net FCN was first proposed in [5] for bio-medical image segmentation. Since then, it has been successfully adapted for multiple application areas such as autonomous driving, microscopy cell counting and single-image depth estimation [49–51]. This architecture comprises a spatial encoding (Contracting) path which extracts coarse feature representations, followed by a spatial decoding (Expansive) path to recover the input image spatial dimensions. The original image undergoes a sequence of downsampling operations to capture spatial information in different spatial resolutions.As in CNNs, each of these downsampling operations is followed by a convolutional layer. At the end of the encoder path, coarse feature representations of the input image are obtained. This representation is passed to the decoder path, which consists of a sequence of spatial upsampling operations to recover the input image size. To preserve fine-grained details throughout the network, skip connections are used from each downsampling layer in the encoder path to its corresponding upsampling layer in the decoder path. These skip connections consist of copying the feature maps from the contracting path and concatenating them to the corresponding feature maps in the expansive path in order to preserve fine-grain spatial details at the final representation. This encoder-decoder structure is presented in Figure 6.

3.3.2 Fully Convolutional Dense Network

The dense network was originally designed as a variation of CNNs for image classification tasks. As CNNs become deeper, the information about the input and the gradients during training can get lost after passing through many layers. Dense networks address this issue with Dense Blocks (DB), which consist of a series of convolutional layers with bypassing connections from each layer to all the following layers within the block. These bypassing connections strengthen feature propagation, encourage feature reuse and allow more efficient gradient propagation during training [52]. A dense block is presented in Figure 7.

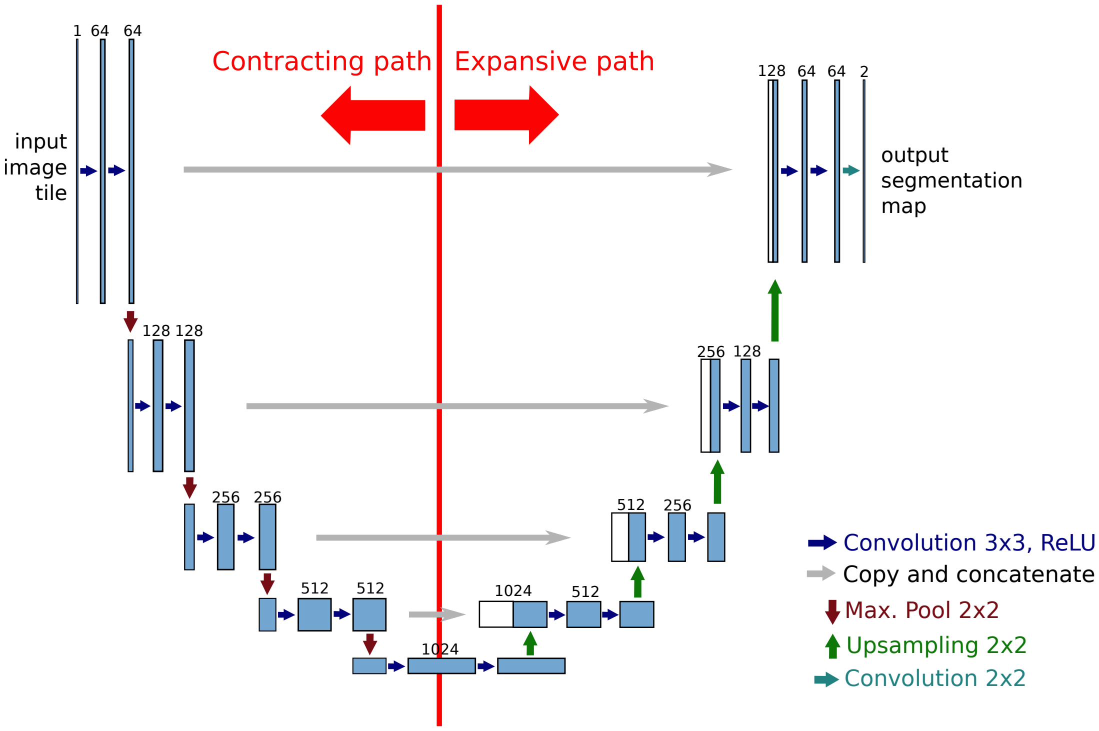

Figure 6: Original U-Net architecture for semantic segmentation. Each blue box corresponds to a multi-channel feature map, where the number of channels is denoted on top of the box. A contractive path applies multiple downsampling operations to extract coarse features, and an expansive path computes upsampling operations to recover the original resolution. Feature maps in the contracting path are copied and concatenated to the expansive path to preserve granular spatial details (Adapted from [5]).

A fully convolutional dense network [6] is a modification of the U-Net, leveraging the concept of *DBs* to obtain a deeper architecture. This network has been recently used in multiple areas such as optical flow prediction and

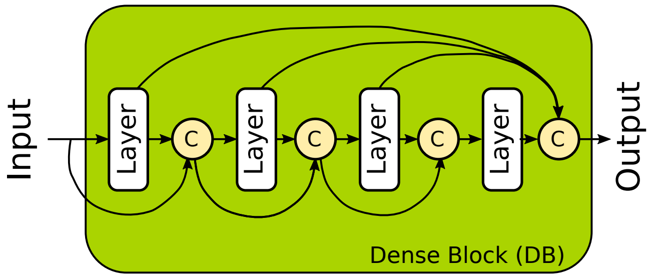

Figure 7: Representation of a Dense Block (*DB*). Input is an image or a feature map with spatial dimensions. A layer consists of a convolution, followed by batch normalization and *ReLU* activation function. Circles represent concatenation (Adapted from [6]).

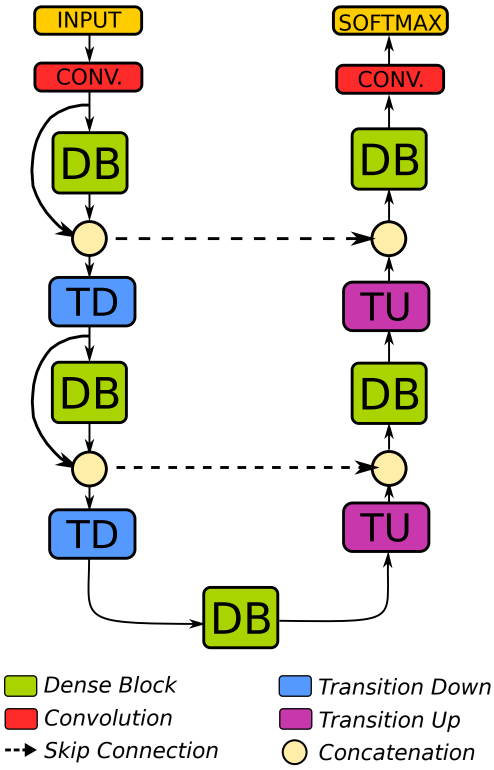

Figure 8: Dense FCN architecture. It consists of a downsampling path with 2 Transition Down (TD) blocks, and an upsampling path with 2 Transition Up (TU) blocks. Circles represent concatenation. Dashed lines represent skip connections, which concatenate feature maps from downsampling stages to the corresponding feature maps from upsampling stages (Adapted from [6]).

medical image segmentation [53, 54]. As the previous approach, it implements a downsampling path which extracts coarse semantic features, followed by an upsampling path responsible for recovering the input spatial resolution in the final output (Figure 8). The downsampling path consists of successive DBs followed by Transition Down (TD) blocks, each of which comprises a convolution and a downsampling operation. Likewise, the upsampling path consists of successive DBs followed by Transition Up (TU) blocks. The TU blocks contain a convolution layer and an upsampling operation. The convolutions in TD and TU blocks allow the network to learn features at multiple spatial scales. Skip connections are added to concatenate feature maps from downsampling stages to the corresponding feature maps in the upsampling stages.

3.3.3 Atrous Spatial Pyramid Pooling (ASPP)

Typical convolutions use small kernel sizes such as 3×3, which consider only local spatial context. In the previous FCN architectures, successive pooling operations inter-leaved with convolutional layers allow to consider a larger spatial context (also known as field of view) without the need to enlarge the convolutional kernel sizes. However, the successive use of downsampling operations reduces the spatial resolution of the resulting feature maps. In the early FCN architectures, this is partially mitigated using deconvolutional layers, but these require additional memory and computation. An alternative method has been recently used for FCNs, replacing the encoder-decoder structures with atrous convolutions [7, 55, 56].

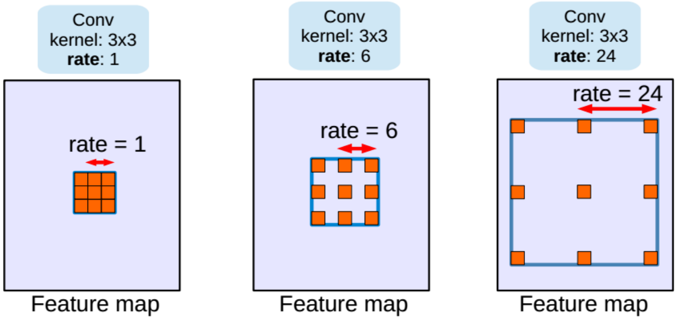

Figure 9: Atrous convolutions overview. A larger field of view is attained by increasing the dilation rate [7].

Instead of reducing the input spatial resolution to consider a larger field of view, atrous convolutions consider a larger spatial context by increasing the convolutional kernel size. This would be inefficient for regular convolutions, because increasing the kernel size would result in a quadratic increase of training parameters. Atrous convolutions solve this problem using a filter with holes, in which a 3×3 filter is upsampled by an *atrous rate* factor, filling with zeros in between filter values. The atrous filter upsampling method is illustrated in Figure 9 for different atrous rate. Note that an atrous convolution with dilation rate 1 is equivalent to a regular convolutional layer.

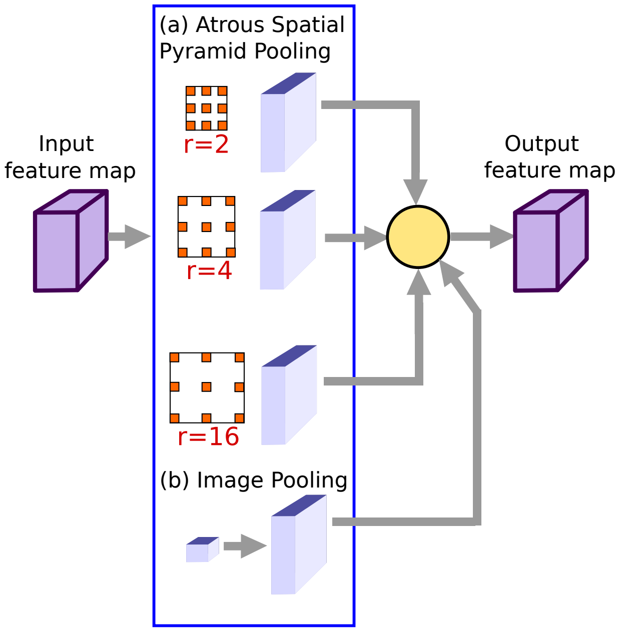

Figure 10: ASPP feature layer. It consists of parallel atrous convolutions with multiple dilation rates. An image pooling layer is also added to consider contextual information from the whole input image (Circle represents concatenation).

A feature layer named Atrous Spatial Pyramid Pooling (ASPP) replaces the use of downsampling and upsampling operations with atrous convolutions for FCN architectures. ASPP comprises a group of atrous convolutional layers with increasing dilation rate values, applied in parallel to the same input tensor. Resulting feature representations from these atrous convolutions are then stacked together to form the ASPP output.

Although atrous convolutions consider spatial information at multiple scales, they can't extract contextual information from the whole input image. Because of this, ASPP additionally incorporates image-level feature extraction with global average pooling (GAP). This operation takes the average of each feature map, resulting in a global representation of size $1 \times 1 \times N_{features}$, where $N_{features}$ is the number of feature representations [57]. Each of these feature maps is then upsampled to the input feature dimensions and the result is concatenated to the ASPP output. The ASPP layer is presented in Figure 10. This feature layer has increasingly been used in multiple application fields [58, 59].

3.4 Recurrent Neural Networks (RNN)

Recurrent Neural Networks (RNN) are a type of neural network designed for processing sequential data. These models are regarded as the state-of-the-art for temporal modeling tasks [60]. RNNs can be seen as neural networks with feedback. Given an input sequence ($\boldsymbol{x} = \boldsymbol{x}_0, \boldsymbol{x}_1, ..., \boldsymbol{x}_T$), the output of such network is given by the equations:

$$\boldsymbol{h}_t = f(\boldsymbol{b} + \boldsymbol{W}\boldsymbol{h}_{t-1} + \boldsymbol{U}\boldsymbol{x}_t) \tag{3-2}$$

$$\boldsymbol{y}_t = g(\boldsymbol{c} + \boldsymbol{V}\boldsymbol{h}_t) \tag{3-3}$$

where $\boldsymbol{h}_t$ is the state at time step t, $\boldsymbol{W}$, $\boldsymbol{U}$ and $\boldsymbol{V}$ are weight matrices, $\boldsymbol{b}$ and $\boldsymbol{c}$ are bias vectors and $\boldsymbol{y}_t$ is the network output for time step t. f and g are activation functions, usually *tanh* and *softmax*, respectively.

Because of their recurrent nature, RNNs can compute a different value at its output for each time step in the input sequence. A many-to-one network considers only one element of the output sequence (e.g. the last of them). In contrast, a many-to-many recurrent network considers the entire output sequence. In the latter case, the training total loss computed by the sum of the losses over all time steps. This configuration is useful for multidate crop recognition because classifications for the entire image sequence can be obtained by a single model. Figure 11 shows on the left the basic RNN architecture and on the right its unrolled representation for three time steps.

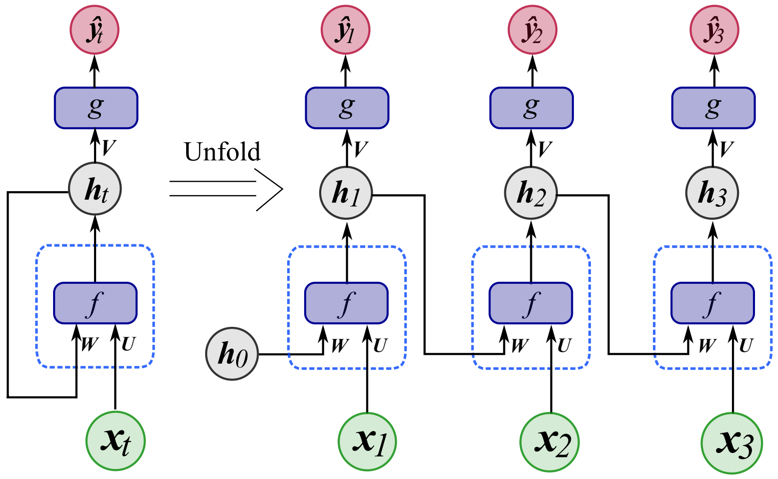

Figure 11: Many-to-many basic RNN.

To produce the outcome $\boldsymbol{x}_t$ at time t the basic RNN relies on the current input $\boldsymbol{x}_t$ and on a summary of prior time steps coded in the previous state $\boldsymbol{h}_{t-1}$. When available, inputs at posterior instants can be used to improve the classification results at time t. This is achieved by bidirectional RNNs. They consist of two RNNs trained simultaneously. The first RNN is trained in the temporal forward direction, whereas the second one is trained in the backward direction [61]. Correspondent state vectors from both RNNs, $\vec{\boldsymbol{h}}_t$ and

$\boldsymbol{h}_t$ are usually concatenated to form the unified state vector $\boldsymbol{h}_t$. This scheme is illustrated in Figure 12 for a sequence of length equal to 3.

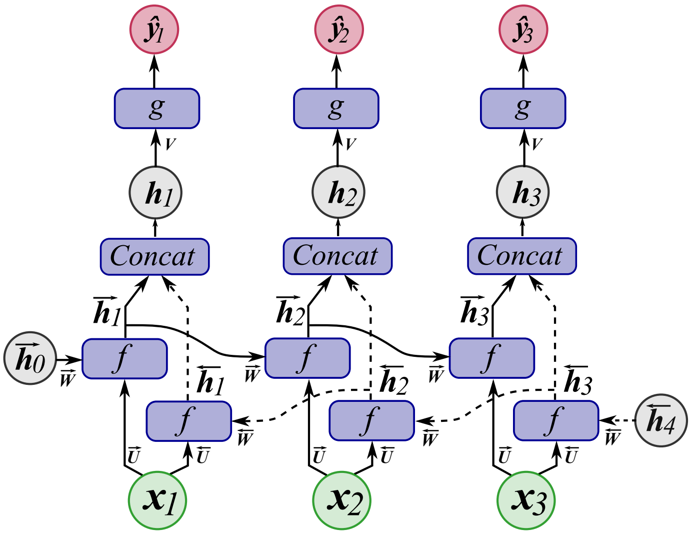

Figure 12: Bidirectional RNN for three time steps (Unfolded representation).

3.5 Long Short Term Memory Networks (LSTM)

LSTMs are a special type of RNN that are capable of modeling both long and short term time dependencies. The main improvement against traditional RNNs is a memory cell $\boldsymbol{C}_t$ which can be accessed, written and cleared by trainable gates (See Figure 13). Specifically, the model uses an information gate $\boldsymbol{i}_t$ to select which information is added to the cell; a forget gate $\boldsymbol{f}_t$ to discard useless previous knowledge and an output gate $\boldsymbol{o}_t$ to produce the final result. In the original architecture, the LSTM internal operations are implemented as fully connected neural network layers.

Given an input sequence $\boldsymbol{x} = \boldsymbol{x}_0, \boldsymbol{x}_1, ..., \boldsymbol{x}_T$, equations for this model are as follows, where "$\circ$" denotes the Hadamard product:

$$
\begin{aligned}
&\boldsymbol{i}_t = \sigma(\boldsymbol{W}_{xi}\boldsymbol{x}_t + \boldsymbol{W}_{hi}\boldsymbol{h}_{t-1} + \boldsymbol{W}_{ci} \circ \boldsymbol{C}_{t-1} + \boldsymbol{b}_i)\\
&\boldsymbol{f}_t = \sigma(W_{xf}\boldsymbol{x}_t + \boldsymbol{W}_{hf}\boldsymbol{h}_{t-1} + \boldsymbol{W}_{cf} \circ \boldsymbol{C}_{t-1} + \boldsymbol{b}_f)\\
&\boldsymbol{C}_t = \boldsymbol{f}_t \circ \boldsymbol{C}_{t-1} + \boldsymbol{i}_t \circ tanh(\boldsymbol{W}_{xc}\boldsymbol{x}_t + \boldsymbol{W}_{hc}\boldsymbol{h}_{t-1} + \boldsymbol{b}_c)\\
&\boldsymbol{o}_t = \sigma(\boldsymbol{W}_{xo}\boldsymbol{x}_t + \boldsymbol{W}_{ho}\boldsymbol{h}_{t-1} + \boldsymbol{W}_{co} \circ \boldsymbol{C}_t + \boldsymbol{b}_o)\\
&\boldsymbol{h}_t = \boldsymbol{o}_t \circ tanh(\boldsymbol{c}_t)
\end{aligned}
$$

In these equations, $\boldsymbol{h}_t$ is the hidden vector at timestep t; $\boldsymbol{W}_{xi}, \boldsymbol{W}_{hi}, \boldsymbol{W}_{ci}, \boldsymbol{W}_{xf}, \boldsymbol{W}_{hf}, \boldsymbol{W}_{cf}, \boldsymbol{W}_{xc}, \boldsymbol{W}_{hc}, \boldsymbol{W}_{xo}, \boldsymbol{W}_{ho}, \boldsymbol{W}_{co}$ are weight matrices and $\boldsymbol{b}_i, \boldsymbol{b}_f, \boldsymbol{b}_c, \boldsymbol{b}_o$ are bias vectors.

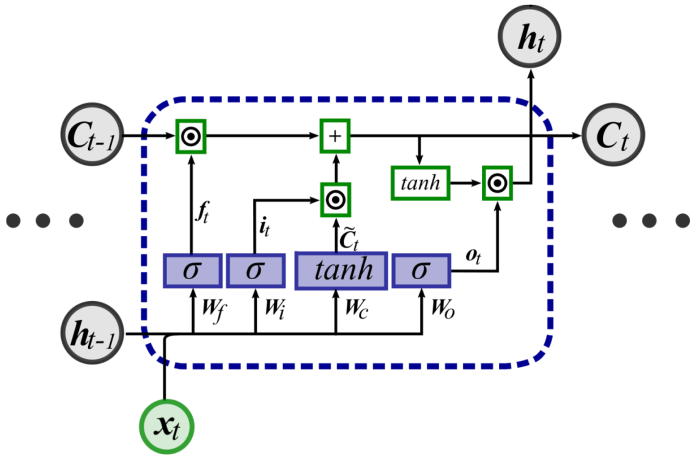

Figure 13: LSTM structure diagram.

3.6 Convolutional Long Short Term Memory Networks (ConvLSTM)

LSTM's major drawback in handling spatial data is the usage of fully connected layers for its input-to-state and state-to-state transitions, which do not take spatial context into account. To overcome this problem, a ConvLSTM cell takes the original LSTM (Figure 13) and replaces the fully connected layers from the forget, information and output gates with convolutional layers . Inputs $\mathbf{x}_1, ...\mathbf{x}_t$, hidden states $\mathbf{h}_1, ...\mathbf{h}_t$ and cell outputs $\mathbf{C}_1, ...\mathbf{C}_t$ are 3D tensors whose first two dimensions are spatial dimensions (rows and columns), and the third dimension corresponds to the number of feature representations [17]. The state equations for a ConvLSTM are as follows, where '$*$' denotes the convolution operator and '$\circ$' denotes the Hadamard product:

$$\begin{aligned}
\boldsymbol{i}_t &= \sigma(\boldsymbol{W}_{xi} * \boldsymbol{\mathcal{X}}_t + \boldsymbol{W}_{hi} * \boldsymbol{\mathcal{H}}_{t-1} + \boldsymbol{W}_{ci} * \boldsymbol{C}_{t-1} + \boldsymbol{b}_i) \\
\boldsymbol{f}_t &= \sigma(\boldsymbol{W}_{xf} * \boldsymbol{\mathcal{X}}_t + \boldsymbol{W}_{hf} * \boldsymbol{\mathcal{H}}_{t-1} + \boldsymbol{W}_{cf} * \boldsymbol{C}_{t-1} + \boldsymbol{b}_f) \\
\boldsymbol{C}_t &= \boldsymbol{f}_t \circ \boldsymbol{C}_{t-1} + \boldsymbol{i}_t \circ tanh(\boldsymbol{W}_{xc} * \boldsymbol{\mathcal{X}}_t + \boldsymbol{W}_{hc} * \boldsymbol{\mathcal{H}}_{t-1} + \boldsymbol{b}_c) \\
\boldsymbol{o}_t &= \sigma(\boldsymbol{W}_{xo} * \boldsymbol{\mathcal{X}}_t + \boldsymbol{W}_{ho} * \boldsymbol{\mathcal{H}}_{t-1} + \boldsymbol{W}_{co} \circ \boldsymbol{C}_t + \boldsymbol{b}_o) \\
\boldsymbol{\mathcal{H}}_t &= \boldsymbol{o}_t \circ tanh(\boldsymbol{C}_t)
\end{aligned}$$

In these equations, $\boldsymbol{\mathcal{X}}_t$ is the input tensor at timestep t and $\boldsymbol{\mathcal{H}}_t$ is the hidden state tensor.

4
RNN ARCHITECTURES FOR MULTIDATE CROP RECOGNITION

In this section we present the recurrent network architectures proposed in this thesis for crop mapping from multitemporal RS data. Firstly, we describe two networks adapted from [18] for many-to-many tasks that served as baseline in our research. Next, the proposed architecture is presented.

4.1
Unidirectional Convolutional LSTM

The first architecture considered in this work is the Unidirectional Convolutional LSTM (UConvLSTM), a unidirectional version of the architecture proposed in [18], which was adapted to many-to-many tasks. Its architecture is shown in Figure 14a. The input sequence goes first through a ConvLSTM net followed by 1×1 convolutions which produces as many activation maps as the number of classes. Next, batch normalization and $ReLU$ activation functions are applied. In the final layer, a $softmax$ function assigns posterior probabilities to each pixel.

4.2
Bidirectional Convolutional LSTM

The second architecture tested in this work is the Bidirectional Convolutional LSTM (BConvLSTM), illustrated in Figure 14b. The BConvLSTM also derives from the architecture proposed in [18] and it was adapted for many-to-many tasks. It can be regarded as a bidirectional version of UConvLSTM, whereby the plain ConvLSTM layer is replaced by a bidirectional ConvLSTM layer. The BConvLSTM network comprises two ConvLSTMs: one processes the input data in the forward direction, while the other operates in reversed, backward direction. The outputs of both ConvLSTM are concatenated to form a single output tensor. From this point on, the architecture does not differ from the previous one. 1×1 convolutions are applied to aforementioned tensor producing one activation map per class, followed by batch normalization and by a $ReLU$ activation function. A *softmax* layer delivers posterior probabilities for each pixel.

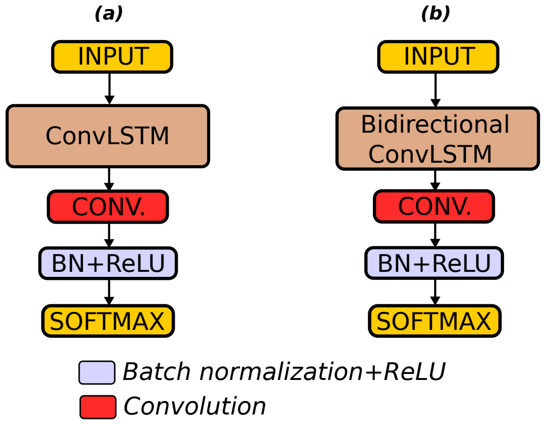

Figure 14: RNN architectures adapted to many-to-many tasks: *(a)* UConvLSTM, *(b)* BConvLSTM. Input is a sequence of images. The output corresponds to a sequence of images with the class probabilities predicted for each image pixel.

4.3 Hybrid Fully Convolutional Recurrent Approaches

Previous approaches use convolutions exclusively at the original input scale. In this case, each output pixel will be classified using only information from a $k \times k$ neighbouring area in the input images, where k is the convolutional kernel size. This limits the network's capabilities to exploit context information at multiple spatial scales. The proposed hybrid networks combine elements of the previous recurrent architectures with the FCN inherent capabilities to perform multi-scale spatial feature extraction. Some of the most relevant FCN architectures were considered for this hybrid design: The U-Net, Dense FCN and ASPP. Thus, three recurrent FCN architectures are proposed: BUnetConvLSTM, BDenseConvLSTM and BAtrousConvLSTM. These architectures are described in the following subsections.

Bidirectional Recurrent U-Net (BUnetConvLSTM)

The BConvLSTM network classifies considering the input spatial and temporal context. However, it lacks the ability to extract features at multiple spatial scales. The proposed BUnetConvLSTM (Figure 15) combines elements of the BConvLSTM architecture with the U-Net fully convolutional network (FCN) from [5], which extracts features at multiple spatial resolutions by successively downsampling the input image dimensions in between convolutional layers. The U-Net FCN comprises a downsampling path, so called encoder, which extracts coarse semantic features, followed by an upsampling path, so called decoder, responsible for recovering the input spatial resolution in the

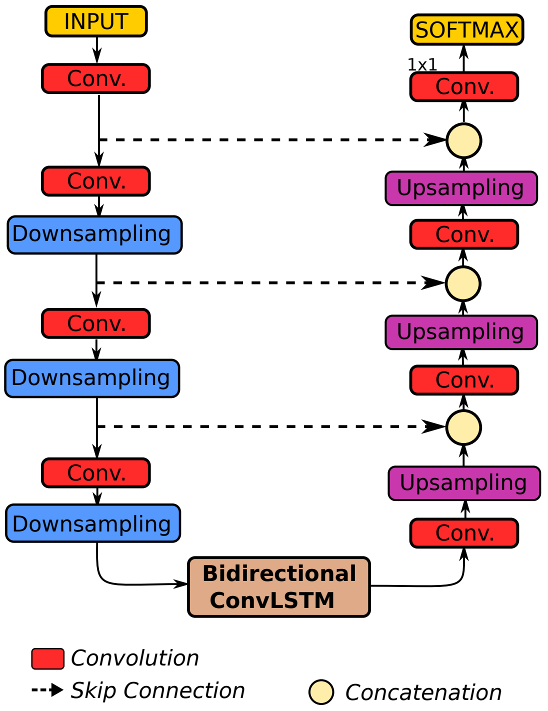

Figure 15: BUnetConvLSTM architecture. Input is a sequence of images. The output corresponds to a sequence of images with the class probabilities predicted for each image pixel.

final output. In the proposed architecture, the input sequence of images is passed through the U-Net encoder to extract coarse features. Then the resulting values are presented to a bidirectional ConvLSTM, which returns the entire sequence of elements at its output. Finally, a decoder is applied to each element in the sequence to recover the spatial resolution from the input images. In this architecture, each element in the downsampling path is formed by a downsampling operation followed by a convolutional layer. Likewise, each element in the upsampling path is formed by an upsampling operation followed by a convolutional layer. Finally a convolution with 1 × 1 kernel size and softmax activation function produces the per-pixel posterior class probabilities.

Bidirectional Recurrent Dense FCN (BDenseConvLSTM)

Similar to the previous approach, the BDenseConvLSTM combines the encoder-decoder structure from the Dense FCN presented in [6] with a bidirectional ConvLSTM network. First, a spatial encoder is applied to the input sequence. This encoder consists of subsequent Dense Blocks followed by Transition Down blocks. Then the resulting sequence of feature representations is passed to a bidirectional ConvLSTM. Finally, a spatial decoder is applied to

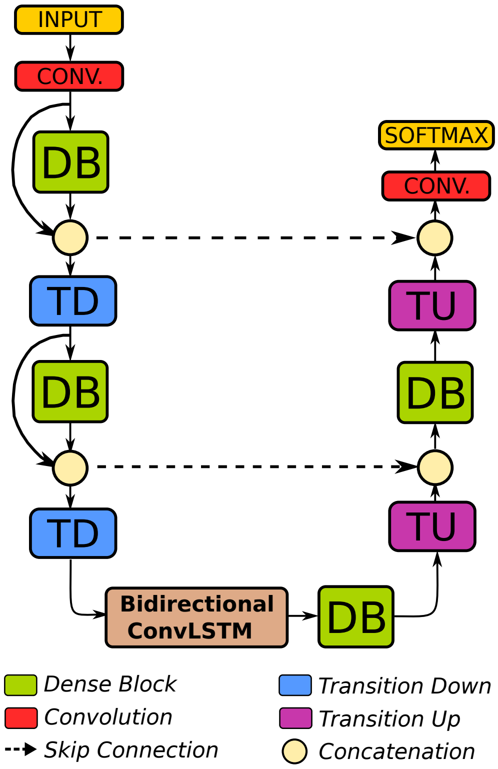

Figure 16: BDenseConvLSTM architecture. The input is a sequence of images. The output corresponds to a sequence of images with the predicted class probabilities for each image pixel.

each element in the sequence to recover the input spatial resolution, which consists of a series of Transition Up blocks followed by Dense Blocks.

Recall that the elements of a Dense FCN network were previously explained in Section 3.3.2. The proposed BDenseConvLSTM architecture is presented in Figure 16. In this architecture, Dense Blocks (DB) are composed of a sequence of convolutional layers with multiple bypassing connections among them. Transition Down (TD) blocks are composed of a convolution and a downsampling operation, while a Transition Up (TU) block performs an upsampling operation. Skip connections are used between downsampling and upsampling stages.

Bidirectional Recurrent ASPP (BAtrousConvLSTM)

The previous hybrid architectures used an encoder-decoder structure to consider spatial context at multiple scales. As an alternative approach, this architecture replaces the encoder-decoder structure with the more recent *Atrous Spatial Pyramid Pooling* (ASPP) module from [55], which uses atrous convolutions to extract features at different spatial scales without the need to use downsampling or upsampling operations. The proposed BAtrousConvL-

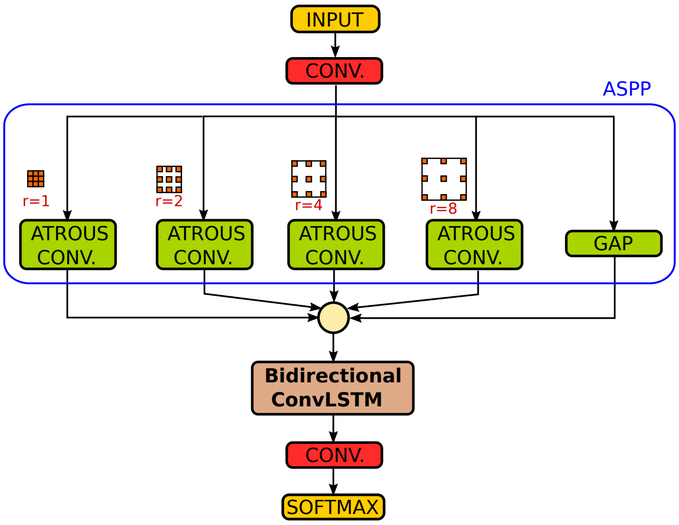

Figure 17: BAtrousConvLSTM architecture. The input is a sequence of images. The output corresponds to a sequence of images with the predicted class probabilities for each image pixel (Circle represents concatenation).

STM architecture combines elements from the BConvLSTM network with the inherent multi-scale feature extraction properties from ASPP.

First, an ASPP module is applied to each of the images in the input sequence. This module uses multiple atrous convolutions in parallel with increasing dilation rates, and additionally extracts global image-level representations with a Global Average Pooling (GAP) layer. The results of these atrous convolutions and GAP layer are then concatenated to form the ASPP output. These features are passed to a bidirectional ConvLSTM to further extract spatio-temporal features, which is configured to return the entire sequence of representations. Finally a convolution with 1 × 1 kernel size and softmax activation function gives the posterior class probabilities for each pixel in the sequence of input images. This architecture is presented in Figure 17.

5
EXPERIMENTAL ANALYSIS

This chapter describes the experiments carried out to validate the methods proposed in the previous chapter. First, the study areas in which the experiments were carried are detailed. Then the experimental protocol is explained, including a description of the hyperparameter configuration used in each network architecture. Finally the results are presented and discussed in terms of average F1 score and *Overall Accuracy* (OA) performance metrics.

5.1
Datasets

Two publicly available datasets for multitemporal crop recognition in tropical regions were used for performance assessment. The first region is located in Campo Verde municipality, Mato Grosso, Brazil, with an extension of 4,782 km^2 [11]. It is located at a latitude of 15°32'48" south and longitude of 55°10'08" west (Figure 18). It features a sequence of 14 pre-processed, dual polarized Synthetic Aperture Radar (SAR) images from Sentinel-1 acquired in the Interferometric Wide Swath, Ground Range Detected Level-1 mode with 250 km swath and 10 m spatial resolution. These images were taken between October 2015 and July 2016, with one or two images per month. However, no image was available in April. The dates corresponding to each image are presented in Table 1. The class distribution greatly varies over time (see Figure 19). *Soybean* is the main crop type from October 2015 to February 2016 and its replaced by *Cotton* and *Maize* in the following months.

The second region is located in Luis Eduardo Magalhães (LEM) municipality, Bahia state, Brazil, with an area of 3,940 km^2 [62]. It is at a latitude of 12°05'31" south and longitude of 45°48'18" west. Its location is also presented in Figure 20. A set of 13 pre-processed Sentinel-1 SAR images obtained between June 2017 and June 2018 was used in our experiments (See Table 2). The images were acquired in the Interferometric Wide Swath, Ground Range Detected Level-1 mode with 10 m spatial resolution. In both cases, the preprocessing step included radiometric and terrain correction, and the VV and VH bands in linear scaling were converted to dB. Similar to Campo Verde, the class distribution in LEM dataset is non uniform along the year, as shown in

Figure 21. The main crop types are *Soybean*, *Maize*, *Cotton* and *Millet*.

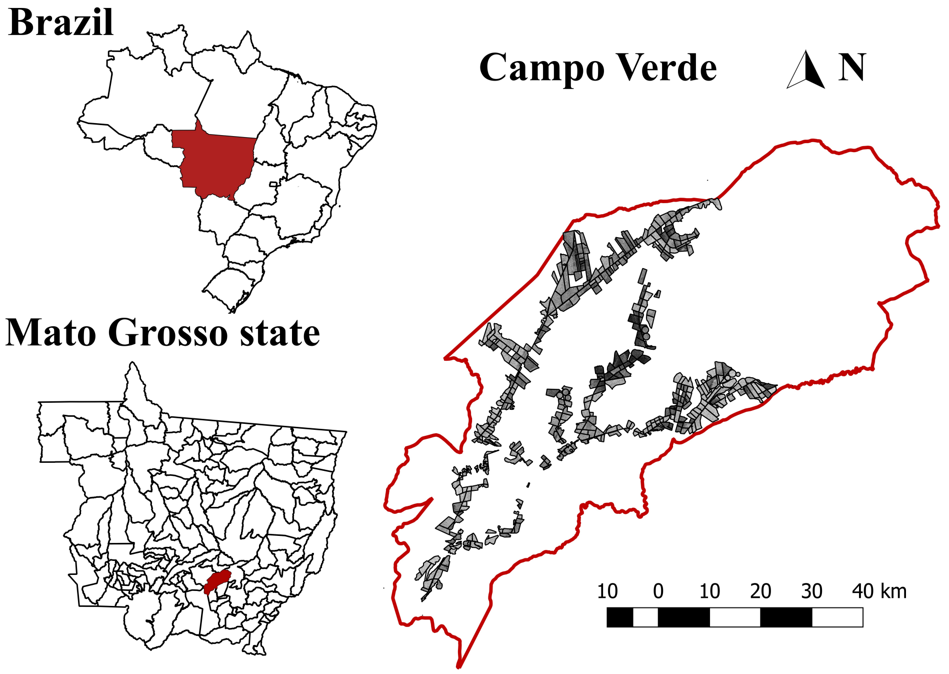

Figure 18: Campo Verde dataset is located in the state of Mato Grosso, Brazil. It comprises 513 parcels with 50% used for training and 50% for testing. Taken from [3].

Table 1: Acquisition dates for Campo Verde dataset. A sequence of 14 images was used.

Year	Month	Date
2015	October	29
	November	10, 22
	December	04, 16
2016	January	21
	February	14
	March	09, 21
	April	-
	May	08, 20
	June	13
	July	07, 31

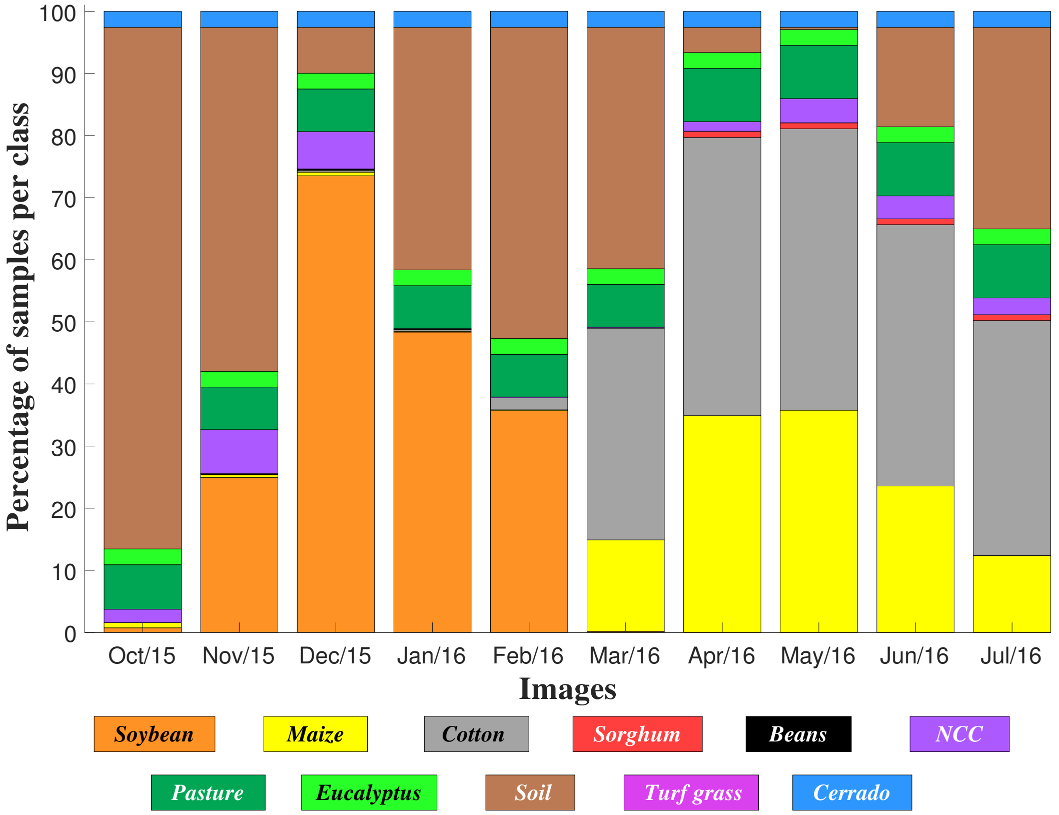

Figure 19: Percentage of classes per date in Campo Verde study area. Taken from [4].

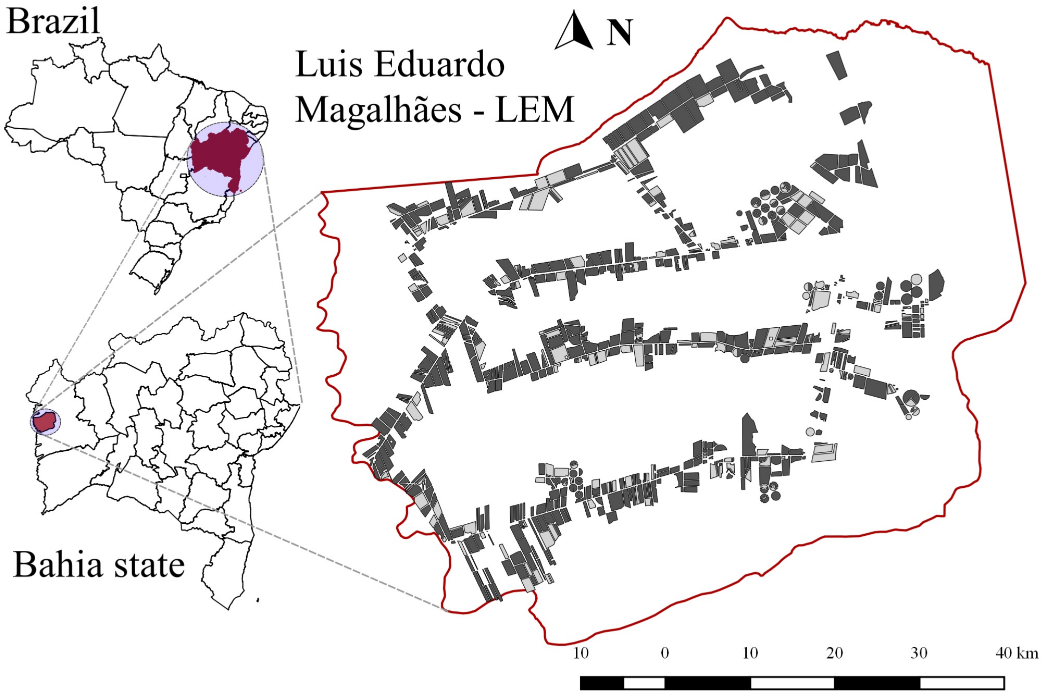

Figure 20: The LEM dataset is located in the state of Bahia, Brazil. It comprises 794 parcels, from which 75% is used for training (Parcels in dark gray) and 25% is used for testing (Parcels in light gray). Taken from [4].

Table 2: Acquisition dates used for LEM dataset. Images from 13 dates were considered.

Year	**Month**	**Date**
2017	June	12
	July	06
	August	11
	September	16
	October	10
	November	15
	December	09
2018	January	14
	February	19
	March	15
	April	20
	May	14
	June	19

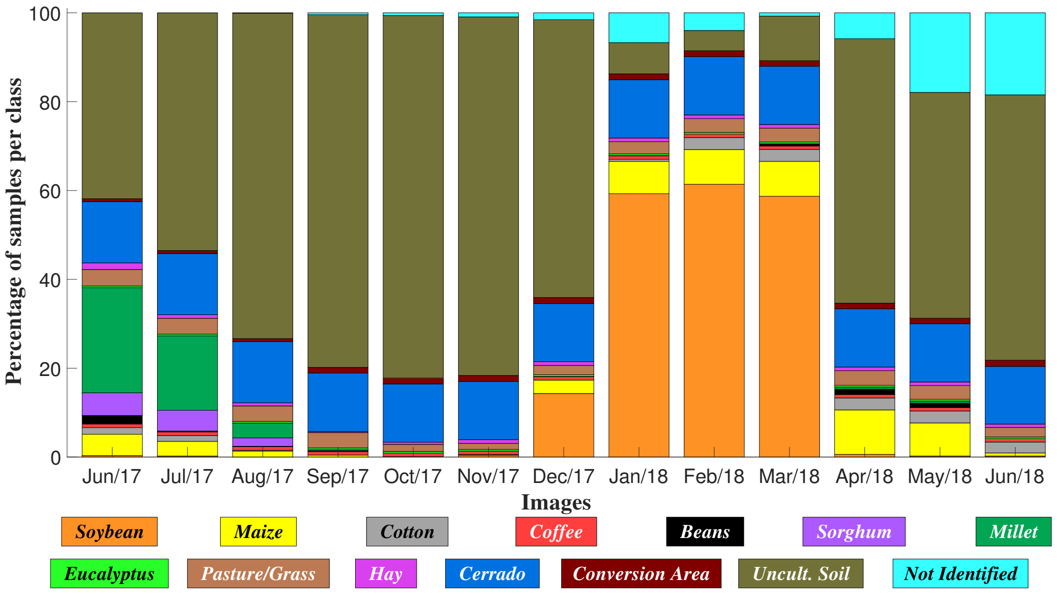

Figure 21: Percentage of classes per date in LEM study area. Taken from [4].

5.2 Experimental Setup

This section describes the experimental setup to test the methods presented in Chapter 4 using the datasets from Section 5.1.

Hyperparameter Configuration

Different hyperparameter values were tested for each method. In this section, the configurations that attained the best results are presented. Parameter setups for UConvLSTM and BConvLSTM networks are shown in Tables 3 and 4, where T represents the temporal sequence length. Following [18], 256 convolutional filters were used in the UConvLSTM network for each LSTM internal gate. Likewise, the BConvLSTM model was configured with 256 recurrent filters per gate: 128 for each direction.

Layer	**Output Shape**	**Filters**
Input	$T \times 32 \times 32$	2
ConvLSTM	$T \times 32 \times 32$	256
Conv.	$T \times 32 \times 32$	$\#classes$

Table 3: UConvLSTM parameter configuration - T is the sequence length

Following [14], the BDenseConvLSTM network was built with two convolutional layers per dense block and 20% as dropout factor. Further details from this architecture are presented in Table 5. Likewise, parameter configuration

Layer	Output Shape	Filters
Input	$T \times 32 \times 32$	2
Bidirectional ConvLSTM	$T \times 32 \times 32$	256
Conv.	$T \times 32 \times 32$	$\#classes$

Table 4: BConvLSTM parameter configuration - T is the sequence length.

for BUnetConvLSTM and BAtrousConvLSTM is shown in Table 6 and Table 7. In BDenseConvLSTM and BUnetConvLSTM, *Average Pooling* was empirically selected as downsampling operator. Except for the last convolution, 3×3 filters were adopted in all cases.

Layer	Output Shape	Filters
Input	$T \times 32 \times 32$	2
DB	$T \times 32 \times 32$	80
Downsampling	$T \times 16 \times 16$	80
DB	$T \times 16 \times 16$	112
Downsampling	$T \times 8 \times 8$	112
Bidirectional ConvLSTM	$T \times 8 \times 8$	256
DB	$T \times 8 \times 8$	32
Upsampling	$T \times 16 \times 16$	144
DB	$T \times 16 \times 16$	32
Upsampling	$T \times 32 \times 32$	112
Conv.	$T \times 32 \times 32$	$\#classes$

Table 5: BDenseConvLSTM parameter configuration - T is the sequence length.

Layer	Output Shape	Filters
Input	$T \times 32 \times 32$	2
Conv.	$T \times 32 \times 32$	16
Downsampling	$T \times 16 \times 16$	16
Downsampling	$T \times 8 \times 8$	32
Downsampling	$T \times 4 \times 4$	64
Bidirectional ConvLSTM	$T \times 4 \times 4$	256
Upsampling	$T \times 8 \times 8$	64
Upsampling	$T \times 16 \times 16$	32
Upsampling	$T \times 32 \times 32$	16
Conv.	$T \times 32 \times 32$	16
Conv.	$T \times 32 \times 32$	$\#classes$

Table 6: BUnetConvLSTM parameter configuration - T is the sequence length.

Layer	**Output Shape**	**Filters**
Input	$T \times 32 \times 32$	2
Conv.	$T \times 32 \times 32$	16
Conv.	$T \times 32 \times 32$	16
ASPP	$T \times 32 \times 32$	320
Bidirectional ConvLSTM	$T \times 32 \times 32$	256
Conv.	$T \times 32 \times 32$	16
Conv.	$T \times 32 \times 32$	$\#classes$

Table 7: BAtrousConvLSTM parameter configuration - T is the sequence length.

Experimental Protocol

Parcels present in the dataset were randomly separated in training and testing sets, whereby the training set contained about 50% of all pixels for Campo Verde and 75% for LEM. These distributions were selected equal to previous works with these datasets for comparison purposes [4, 14]. In each study area, the image was split into non-overlapping image patches to be independently processed by the network. Following [14], the spatial dimensions for the image patch size were selected as 32×32 pixels, corresponding to an area of $320 \times 320m$. Thus, the input patch shape was $14 \times 32 \times 32$ pixels for Campo Verde and $11 \times 32 \times 32$ for LEM. In Campo Verde, 4988 image patches were used for training and 4671 for testing. For LEM, 7420 image patches were used for testing and 2562 image patches for testing. After training, the patch-wise classification results for the test areas were arranged in a mosaic for the final output.

Data augmentation strategies such as rotation, horizontal and vertical flip were used, since they were empirically found to improve overall and per-class performance metrics. Early stopping criteria was used to avoid overfitting, with patience of 10 epochs. In all cases, Adagrad optimizer with learning rate 0.01 was used following [14]. Weighted categorical cross entropy function was used to further compensate the class imbalance inherent in both datasets. Experiments were carried out using Keras framework with Tensorflow backend, on a NVIDIA GTX Titan GPU. The code of these architectures is available upon request.

Results are presented in a per-month basis. In the case of Campo Verde dataset, where 1 or 2 images per month are available, the image with the latest date was selected for analysis.

5.3 Results

Campo Verde Dataset

The results on Campo Verde dataset in terms of overall acuracy (OA) are shown in Figure 22. The basic UConvLSTM approach (blueish bars) presented lower scores compared to its bidirectional counterpart BConvLSTM (greenish bars) for all dates, with a larger difference for the early dates. This occurred because UConvLSTM doesn't take enough multi temporal information into account for the first dates.

BDenseConvLSTM (redish bars) and BUnetConvLSTM (cyanish bars) displayed higher OA scores for all dates compared to BConvLSTM (greenish bars). This indicates that adding spatial encoding and decoding layers allowed the recurrent network to handle a more compact and discriminative representation over the sequence.

The BAtrousConvLSTM (purplish bars) obtained higher OA values than the BConvLSTM. In terms of OA, it performed similar to the other encoder-decoder approaches. This indicates that the atrous spatial pooling module is a valid approach for multi-scale feature extraction without the need of image downsampling stages.

The months with higher OA values were October and December. This could be explained because in these months, more than 70% of the image corresponds to a single class. In October, most of the areas correspond to Soil class while Soybean class is predominant in December. In these months, the network only needed to predict most pixels with the predominant class to obtain a high OA score. However, December is also a month with high F1 score, meaning that the overall classification results were the most appropriate in this month according to the studied metrics.

Per-date average F1 scores for Campo Verde are shown in Figure 23. Compared to the remaining approaches, UConvLSTM performance (blueish bars) was significantly low for the first dates, with 10% F1 score for the first month. Then its performance gradually increased, achieving similar scores to BConvLSTM (greenish bars) for the last months of May, June and July. Encoder-decoder methods (reddish and cyanish bars) and BAtrousConvLSTM (purplish bars) outperformed BConvLSTM (greenish bars) in terms of F1 score, with BAtrousConvLSTM (purplish bars) achieving slightly higher values compared to BDenseConvLSTM and BUnetConvLSTM (reddish and cyanish bars, respectively).

Another method to assess the networks performance is the per-class F1 score. Table 8 shows the F1 scores for the most representative crop types in

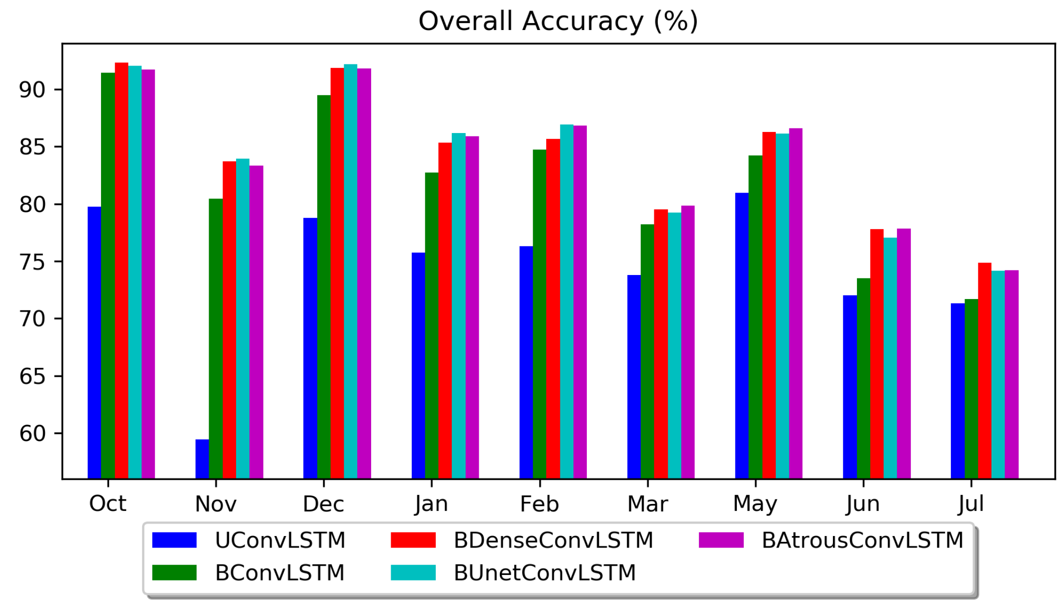

Figure 22: Overall Accuracy for Campo Verde study area, computed in each date.

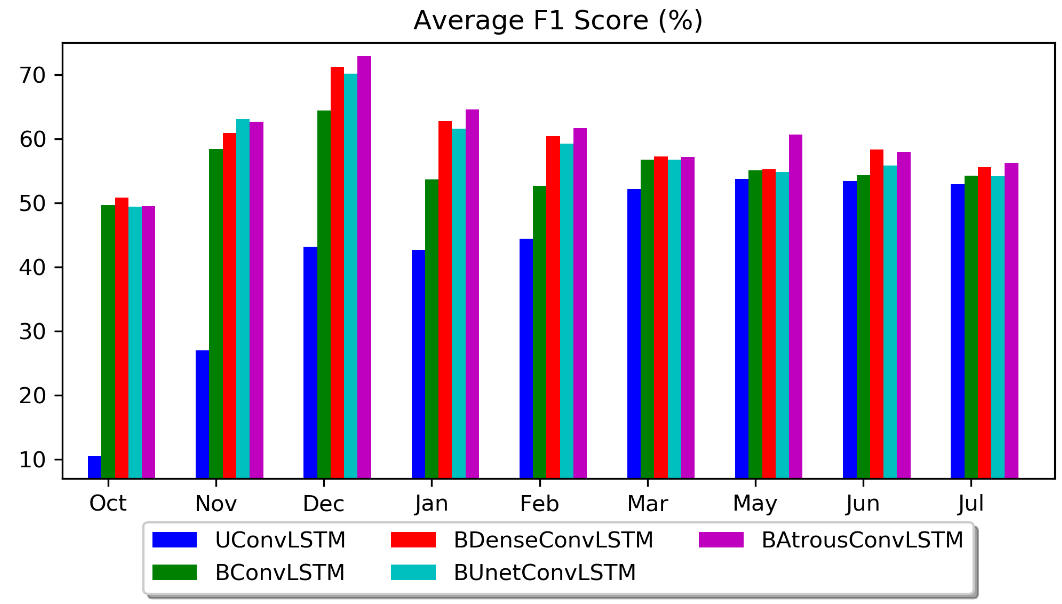

Figure 23: Average F1-Score for Campo Verde study area, computed in each date.

Campo Verde, with the best results highlighted in bold. With few exceptions, the best performance was achieved by the models that use multi-scale feature extraction: BDenseConvLSTM, BUnetConvLSTM and BAtrousConvLSTM. This can be qualitatively assessed in Figure 26 that presents spatial results for sample test areas. In this figure, the UConvLSTM presented a large amount

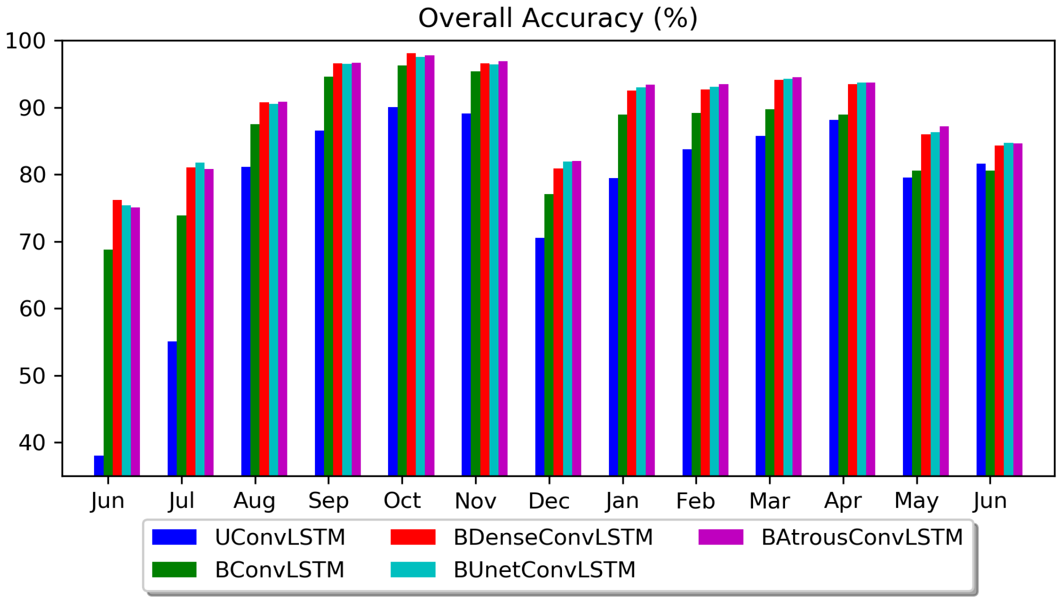

Figure 24: Overall Accuracy for LEM study area, computed in each date.

of errors for the first dates compared to BConvLSTM. Then for the latter date May 2015, results from UConvLSTM were close to its bidirectional counterpart. The UConvLSTM and BConvLSTM architectures presented larger amounts of salt and pepper noise compared to the proposed architectures. This might have occured because the UConvLSTM and BConvLSTM compute convolutions at the input image resolution only, with a fixed kernel size of 3×3. This means that every pixel was predicted using the spatial information of a 3×3 neighborhood, which ignores the images spatial context at larger scales. Instead, the hybrid approaches are designed to extract information from the images at multiple spatial scales, either by successively downsampling the image size (As in BUnetConvLSTM and BDenseConvLSTM) or increasing the convolution kernel size (As in BAtrousConvLSTM). Because of this, the hybrid networks presented smoother classification maps compared to the basic UConvLSTM and BConvLSTM.

The Eucalyptus class obtained F1 scores above 90% for all dates in the hybrid networks. This might be because the parcels corresponding to this crop type remain in the same locations across the entire temporal series, which might make it easier for the network to correctly detect it.

Given the similar performance of the three proposed hybrid networks, further in this document an analysis in terms of training and inference time is presented for better understanding of their differences.

LEM Dataset

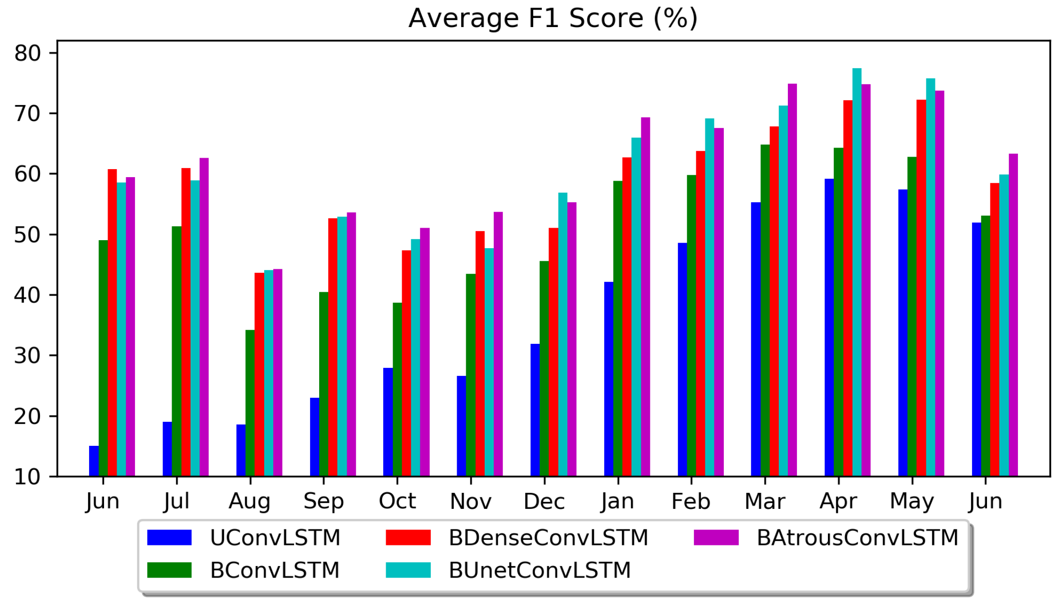

Figure 25: Average F1-Score for LEM study area, computed in each date.

Figure 24 shows per-date OA for LEM. Consistent with previous results, UConvLSTM (blueish bar) presented the lowest performance compared to the other approaches. BConvLSTM (greenish bar) outperformed UConvLSTM in about 28% in the first month and 5% in March. However, UConvLSTM approached BConvLSTM more and more in subsequent months. These values indicate the importance of exploiting future and past multi-temporal information, as in the bidirectional methods.The BDenseConvLSTM, BUnetConvLSTM and BAtrousConvLSTM models (reddish, cyanish and purplish bars, respectively) achieved higher metrics compared to BConvLSTM and UConvLSTM (blueish and greenish, respectively), indicating the importance of their multi-scale feature extraction capability.

The per-date average F1 scores in LEM is presented in Figure 25 lead to similar conclusions. In UConvLSTM (blueish bars), the F1 score started at a low value and gradually increased, although it only reached a value similar to BConvLSTM (greenish bars) for the last date, indicating that future multi-temporal information was useful also in terms of this metric. BConvLSTM (greenish bars) achieved a per-date F1 average score of 50%, whereas UConvLSTM (reddish bars) stayed at 25% from UConvLSTM. The BDenseConvLSTM, BUnetConvLSTM and BAtrousConvLSTM networks achieved the highest scores with 60, 61 and 62% respectively.

Per-class F1 score for LEM study area are presented in Table 9 and Table 10. Clearly, the multi-scale extracting models obtained the best performance across all classes. This could have occurred because of these networks' properties for taking larger spatial context information into account. Spatialized results for LEM dataset are presented in Figure 27 for qualitative assessment. In the figure, BConvLSTM significantly outperformed UConvLSTM while BDenseConvLSTM, BUnetConvLSTM and BAtrousConvLSTM produced less noisy results, reducing the salt and pepper effect observed in the results of BConvLSTM network. This is also due to their capability of exploiting information at multi-scales.

Inference and training time

Another aspect to address in an analyze of the proposed networks is the inference and training times. Lower inference times might be important for real-world applications. Inference times measured in the experiments on Campo Verde test areas are presented in Table 11. The proposed BUnetConvLSTM presented a significantly lower inference time compared to the basic UConvLSTM and BConvLSTM. This occurred because the computational load associated to the convolutional layers are directly related to the amount of data being processed. In BUnetConvLSTM, the input image is successively downsampled during the encoding stage, reducing the computational complexity. In contrast, the UConvLSTM and BConvLSTM perform all the computations at the original image size, which results in larger computational complexity.

BDenseConvLSTM involved a longer inference time compared to BUnetConvLSTM. This might be because the added complexity of the BDenseConvLSTM architecture. Finally, the BAtrousConvLSTM presented the highest inference time. This could be because convolutions are applied at the original input spatial resolution which results in larger computational costs. These inference times could be improved by applying further software optimizations which weren't taken into account during this work.

Table 11: Inference times for the proposed network architectures.

Network	**Inference time [s]**	
	Campo Verde	**LEM**
UConvLSTM	23.9	15.5
BConvLSTM	22.4	12.2
BUnetConvLSTM	12	7.1
BDenseConvLSTM	22.5	13.5
BAtrousConvLSTM	42.7	22.7

	Crop	Month (%)								
	Type	**Oct**	**Nov**	**Dec**	**Jan**	**Feb**	**Mar**	**May**	**Jun**	**Jul**
UConvLSTM	Soybean	0.0	57.6	90.4	82.0	76.5	**39.2**	-	-	-
	Maize	0.0	0.0	10.9	15.2	26.2	54.4	81.6	64.2	**45.9**
	Cotton	-	-	47.0	52.3	27.7	77.1	89.1	87.8	85.2
	Sorghum	-	-	0.2	0.5	2.8	8.4	50.3	50.9	**53.1**
	Beans	-	15.4	39.1	-	-	-	36.2	-	-
	Eucalyptus	4.9	39.5	61.8	70.2	75.3	81.2	83.8	84.7	86.0
BConvLSTM	Soybean	27.0	74.5	96.6	85.8	84.5	37.3	-	-	-
	Maize	44.3	73.5	57.5	0.6	3.0	70.1	87.3	66.1	42.1
	Cotton	-	-	73.2	71.4	43.7	80.2	91.8	89.1	86.1
	Sorghum	-	-	14.7	13.4	12.3	11.8	50.5	49.8	50.4
	Beans	-	28.3	29.8	-	-	-	33.9	-	-
	Eucalyptus	95.3	94.4	93.4	93.1	93.2	89.1	85.8	85.6	86.3
BDenseConvLSTM	Soybean	32.9	78.7	98.2	88.4	86.0	37.7	-	-	-
	Maize	**68.3**	**89.1**	80.9	64.5	**71.0**	72.8	**90.3**	**72.8**	43.6
	Cotton	-	-	75.0	**78.0**	46.1	**81.6**	**92.6**	90.6	**87.6**
	Sorghum	-	-	**38.3**	**44.4**	**31.3**	**18.0**	**52.4**	**53.1**	50.0
	Beans	-	41.9	60.6	-	-	-	35.5	-	-
	Eucalyptus	95.5	95.1	95.1	94.5	92.9	93.3	92.4	92.6	92.4
BUnetConvLSTM	Soybean	34.7	**79.4**	**98.3**	**88.7**	86.6	36.0	-	-	-
	Maize	56.4	84.9	80.0	59.3	44.1	**73.8**	89.6	71.2	43.1
	Cotton	-	-	65.9	73.5	**58.7**	79.8	92.0	89.7	86.9
	Sorghum	-	-	27.9	30.1	21.8	15.4	43.5	43.3	39.7
	Beans	-	**64.8**	**73.7**	-	-	-	40.3	-	-
	Eucalyptus	96.4	96.6	96.4	**96.4**	95.8	**95.8**	**95.0**	**94.7**	**93.8**
BAtrousConvLSTM	Soybean	**40.0**	**79.4**	98.1	88.2	**86.7**	31.0	-	-	-
	Maize	56.1	86.8	**86.8**	**75.4**	65.2	72.6	90.0	71.7	44.5
	Cotton	-	-	**75.7**	70.6	55.6	80.8	92.3	**90.7**	**87.6**
	Sorghum	-	-	27.4	27.6	22.3	13.0	50.8	50.2	47.3
	Beans	-	43.9	54.9	-	-	-	**62.7**	-	-
	Eucalyptus	**97.2**	**97.0**	**96.9**	96.2	**96.1**	95.3	94.9	93.0	93.1

Table 8: Average F1 score for the most relevant crop types in Campo Verde study area, computed at each date from October 2015 to July 2016

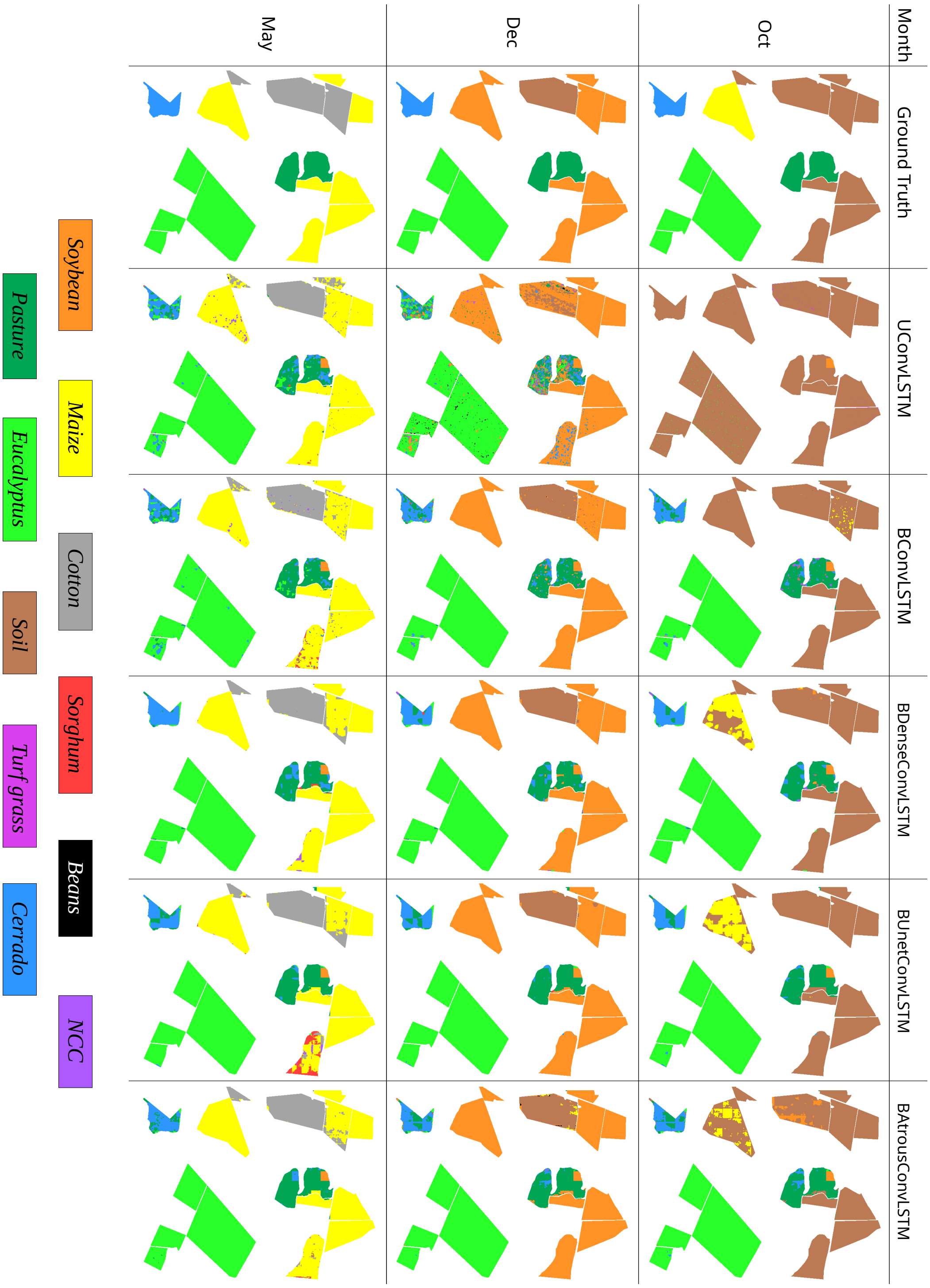

Figure 26: Sample structured output for Campo Verde.

	Crop Type	Month (%)								
		Oct	**Nov**	**Dec**	**Jan**	**Feb**	**Mar**	**May**	**Jun**	**Jul**
UConvLSTM	Soybean	0.0	57.6	90.4	82.0	76.5	**39.2**	-	-	-
	Maize	0.0	0.0	10.9	15.2	26.2	54.4	81.6	64.2	**45.9**
	Cotton	-	-	47.0	52.3	27.7	77.1	89.1	87.8	85.2
	Sorghum	-	-	0.2	0.5	2.8	8.4	50.3	50.9	**53.1**
	Beans	-	15.4	39.1	-	-	-	36.2	-	-
	Eucalyptus	4.9	39.5	61.8	70.2	75.3	81.2	83.8	84.7	86.0
BConvLSTM	Soybean	27.0	74.5	96.6	85.8	84.5	37.3	-	-	-
	Maize	44.3	73.5	57.5	0.6	3.0	70.1	87.3	66.1	42.1
	Cotton	-	-	73.2	71.4	43.7	80.2	91.8	89.1	86.1
	Sorghum	-	-	14.7	13.4	12.3	11.8	50.5	49.8	50.4
	Beans	-	28.3	29.8	-	-	-	33.9	-	-
	Eucalyptus	95.3	94.4	93.4	93.1	93.2	89.1	85.8	85.6	86.3
BDenseConvLSTM	Soybean	32.9	78.7	98.2	88.4	86.0	37.7	-	-	-
	Maize	**68.3**	**89.1**	80.9	64.5	**71.0**	72.8	**90.3**	**72.8**	43.6
	Cotton	-	-	75.0	**78.0**	46.1	**81.6**	**92.6**	90.6	**87.6**
	Sorghum	-	-	**38.3**	**44.4**	**31.3**	**18.0**	**52.4**	**53.1**	50.0
	Beans	-	41.9	60.6	-	-	-	35.5	-	-
	Eucalyptus	95.5	95.1	95.1	94.5	92.9	93.3	92.4	92.6	92.4
BUnetConvLSTM	Soybean	34.7	**79.4**	**98.3**	**88.7**	86.6	36.0	-	-	-
	Maize	56.4	84.9	80.0	59.3	44.1	**73.8**	89.6	71.2	43.1
	Cotton	-	-	65.9	73.5	**58.7**	79.8	92.0	89.7	86.9
	Sorghum	-	-	27.9	30.1	21.8	15.4	43.5	43.3	39.7
	Beans	-	**64.8**	**73.7**	-	-	-	40.3	-	-
	Eucalyptus	96.4	96.6	96.4	**96.4**	95.8	**95.8**	**95.0**	**94.7**	**93.8**
BAtrousConvLSTM	Soybean	**40.0**	**79.4**	98.1	88.2	**86.7**	31.0	-	-	-
	Maize	56.1	86.8	**86.8**	**75.4**	65.2	72.6	90.0	71.7	44.5
	Cotton	-	-	**75.7**	70.6	55.6	80.8	92.3	**90.7**	**87.6**
	Sorghum	-	-	27.4	27.6	22.3	13.0	50.8	50.2	47.3
	Beans	-	43.9	54.9	-	-	-	**62.7**	-	-
	Eucalyptus	**97.2**	**97.0**	**96.9**	96.2	**96.1**	95.3	94.9	93.0	93.1

Table 9: Average F1 score for the most relevant crop types in LEM study area, computed at each date. The sequence starts in June 2017 and finishes in June 2018 (Part 1: From June 2017 to December 2017)

	Crop	**Month (%)**					
	Type	**Jan**	**Feb**	**Mar**	**Apr**	**May**	**Jun**
UConvLSTM	Soybean	88.0	91.4	91.5	58.6	74.2	77.2
	Maize	63.5	64.3	64.9	73.7	62.0	35.8
	Cotton	29.6	69.4	80.8	95.9	98.0	97.8
	Coffee	39.6	42.4	45.4	48.6	50.6	51.5
	Beans	-	-	63.1	48.4	43.0	-
	Sorghum	-	-	-	-	-	-
	Millet	-	0.0	0.0	11.7	16.1	0.0
	Eucalyptus	27.8	28.3	28.6	28.7	28.7	28.2
BConvLSTM	Soybean	94.1	95.0	94.4	60.6	83.4	84.5
	Maize	80.3	75.0	72.2	75.3	61.6	25.9
	Cotton	77.8	99.3	98.9	97.7	98.7	97.5
	Coffee	64.0	62.8	64.3	65.5	68.3	70.1
	Beans	-	-	**85.7**	69.6	59.3	-
	Sorghum	-	-	-	-	-	-
	Millet	-	0.0	0.0	27.1	20.9	0.0
	Eucalyptus	33.8	32.5	31.6	31.0	30.7	29.7
BDenseConvLSTM	Soybean	96.1	96.4	96.6	65.4	**88.9**	**88.1**
	Maize	90.6	87.0	**86.8**	86.9	74.6	41.6
	Cotton	**80.5**	99.7	99.6	99.4	**99.8**	**99.8**
	Coffee	85.8	88.7	89.5	**89.3**	**89.8**	**89.6**
	Beans	**-**	**-**	79.8	**77.8**	77.7	**-**
	Sorghum	**-**	**-**	**-**	**-**	**-**	**-**
	Millet	**-**	2.3	9.6	55.7	49.1	0.1
	Eucalyptus	**64.7**	63.2	63.8	**65.5**	66.1	**62.1**
BUnetConvLSTM	Soybean	96.4	96.8	96.8	**70.7**	85.2	**88.1**
	Maize	**91.9**	**87.6**	86.0	88.7	73.6	42.4
	Cotton	70.8	**99.8**	**99.7**	99.5	99.6	99.7
	Coffee	**86.7**	86.3	86.2	88.5	89.3	86.9
	Beans	-	-	77.9	71.2	76.9	-
	Sorghum	-	-	-	-	-	-
	Millet	-	0.4	14.0	**64.5**	**65.1**	0.1
	Eucalyptus	54.7	56.4	57.4	57.4	58.3	54.6
BAtrousConvLSTM	Soybean	**96.7**	**97.2**	**97.1**	48.5	87.4	86.4
	Maize	91.5	87.2	86.5	**90.0**	**76.8**	**47.5**
	Cotton	72.8	**99.8**	**99.7**	**99.6**	99.5	99.0
	Coffee	86.0	**89.8**	**90.4**	88.7	87.0	85.2
	Beans	-	-	79.5	76.2	**80.0**	-
	Sorghum	-	-	-	-	-	-
	Millet	-	**2.9**	**15.2**	51.2	50.3	**13.5**
	Eucalyptus	58.2	**64.5**	**65.7**	61.8	61.5	58.8

Table 10: Average F1 score for the most relevant crop types in LEM study area, computed at each date. The sequence starts in June 2017 and finishes in June 2018 (Part 2: From January 2018 to June 2018)

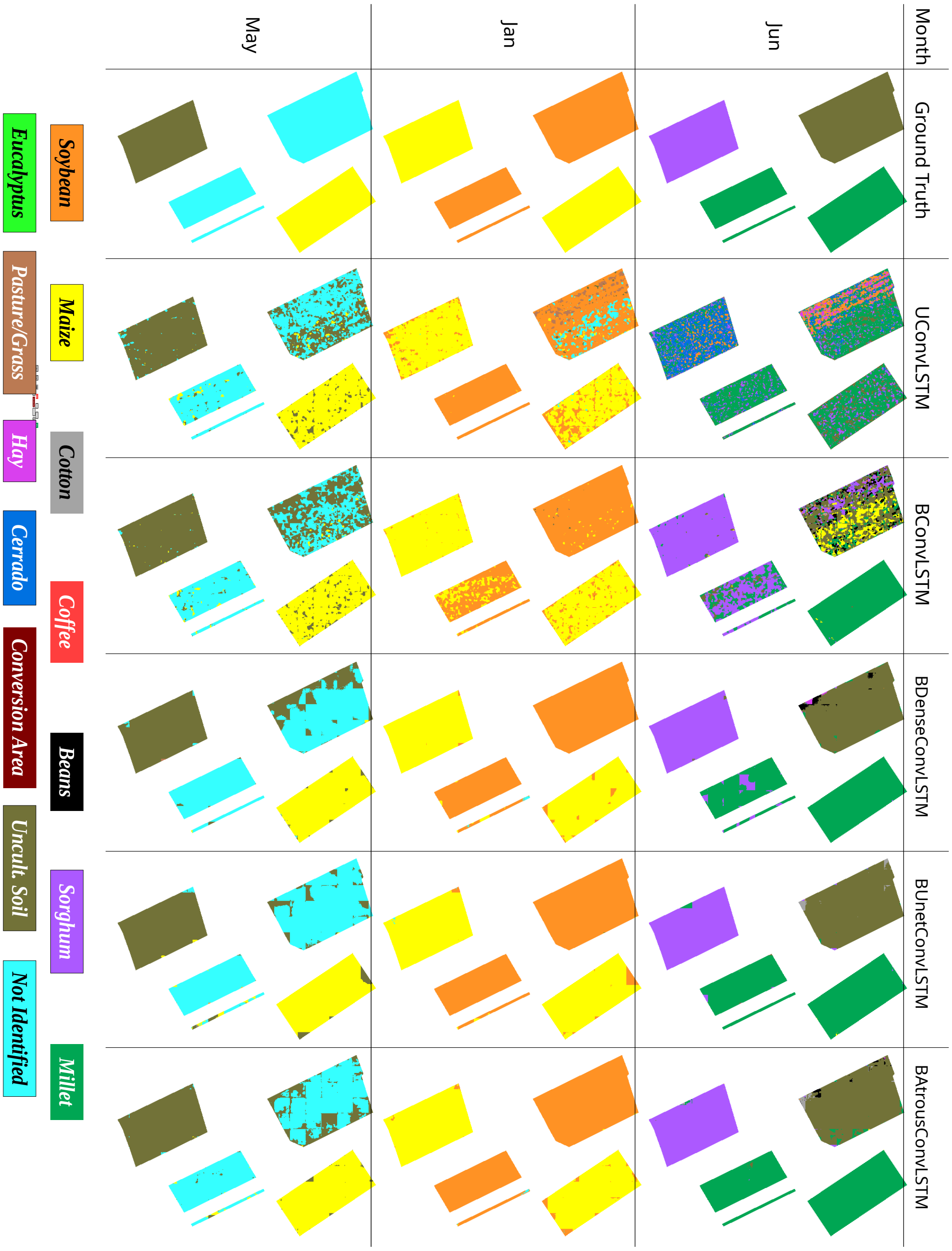

Figure 27: Sample structured output for LEM.

Table 12 presents training times for the proposed networks. In general, the BUnetConvLSTM obtained the lowest training times among all approaches. This might be due to its downsampling stages which reduce the computational cost. Even so, the BDenseConvLSTM required larger training times compared to BUnetConvLSTM and the basic approaches on both datasets, which is likely related to the deeper architecture from BDenseConvLSTM due to its structure based upon dense blocks. The model with larger training time was BAtrousConvLSTM. As with inference times, this high value could be because it applies convolutions at the original spatial resolution, which is computationally expensive. The time measurements were estimated on an equipment with an NVIDIA RTX 2080 Ti GPU.

Table 12: Training times for the proposed network architectures.

Network	**Train time [Hours]**	
	Campo Verde	**LEM**
UConvLSTM	6.76	5.37
BConvLSTM	8.48	4.74
BUnetConvLSTM	4.59	3.87
BDenseConvLSTM	6.5	5.02
BAtrousConvLSTM	8.66	7.04

6 CONCLUSIONS

This work introduced an extension of the traditional ConvLSTM networks for multitemporal crop recognition. In contrast to existing similar approaches, which assign to image sites a single crop type per season, the proposed networks are able to classify crops at each date in the sequence, in the so called many-to-many configuration.

Furthermore, three novel fully convolutional bidirectional recurrent networks called BUnetConvLSTM, BDenseConvLSTM and BAtrousConvLSTM were proposed. These networks use a hybrid approach which combines the multi-scale spatial feature extraction capabilities of FCNs with the spatio-temporal modelling properties of ConvLSTM networks. In particular, BUnetConvLSTM comprises a spatial encoding path to extract coarse features, followed by a ConvLSTM network to further extract spatio-temporal information. Then a spatial decoding path recovers the input spatial resolution for the final pixel-wise predictions. The BDenseConvLSTM is a variation of BUnetConvLSTM which allows to use a deeper architecture by leveraging *Dense Blocks* in its contracting and expanding paths. BAtrousConvLSTM is an alternative to the previous approaches which uses atrous convolutions to extract multi-scale spatial information without the need of downsampling or upsampling stages.

The networks were validated upon two public datasets of tropical regions characterized by highly complex crop dynamics.

In all cases, the bidirectional networks outperformed the unidirectional approach for the first elements of the temporal sequence. This result emphasizes the superiority of bidirectional recurrent networks variants over the unidirectional counterparts in the target application.

The UConvLSTM and BConvLSTM networks produced a salt and pepper effect at their outputs. In contrast, the proposed hybrid approaches, which include an additional spatial encoding stage, reduced this effect and produced smoother predictions at higher accuracy. Thus, the experiments indicated the effectiveness of these convolutional recurrent architectures in exploiting information at multiple spatial scales, improving upon the state of the art architecture from [18] for recurrent approaches in multi-temporal crop recognition. Given that all hybrid architectures presented similar performance metrics, a comparison was made in terms of training and inference times. Amongst the hybrid architectures, the BUnetConvLSTM presented lower processing times compared to its counterparts. Because of this, the BUnetConvLSTM resulted in the most cost-effective alternative across the evaluated approaches.

Future works will focus in making the proposed approaches more operational and usable in a real-world application, by training the networks on multiple datasets and using additional meta information as input to the network such as the date from each image. Besides, attention models could be researched to enhance the recurrent capabilities of the proposed networks. Although SAR data has multiple advantages over other sources of information, the network performance could be further improved by aggregating additional data. Therefore, an adaptation of the proposed networks for data fusion between SAR and other remote sensing sensors such as optical are research directions worth being investigated. Likewise, meteorological data has been used for crop recognition and its aggregation could be useful to further improve the prediction results. Besides, this information is available online [63, 64]. These data fusion approaches could provide higher performance metrics due to the added information while maintaining the robustness to atmospheric obstructions provided by SAR.

Although SAR data provides multiple advantages, it's usage is also challenging because its signal is a function of surface roughness and dielectric constant, largely depending on soil moisture. This makes it difficult for the networks trained in one study area to generalize to other unseen areas. Because of this, future works should focus on evaluating the generalization capabilities of the proposed approach to unseen area. Furthermore, the proposed networks were evaluated in each dataset separately. Future works should also focus on evaluating the inter-dataset network performance and consider training the architectures in multiple datasets to further improve generalization.

Finally, the spatio-temporal modeling capabilities of the proposed approach could be applied to other related problems with spatio-temporal dependencies in diverse areas such as environmental monitoring and petroleum leakage detection.

References

1 RICHARDS, J. A.. **Remote Sensing Digital Image Analysis: An Introduction**. Springer Science & Business Media, 2012.

2 VYAS, K.; SHAH, P.; PATEL, U.; ZAVERI, T. ; OTHERS. **Oil spill detection from sar image data for remote monitoring of marine pollution using light weight imagej implementation**. In: 2015 5TH NIRMA UNIVERSITY INTERNATIONAL CONFERENCE ON ENGINEERING (NUICONE), p. 1–6. IEEE, 2015.

3 LA ROSA, L. E.. **Crop Recognition from Multitemporal SAR Image Sequence Using Deep Learning Techniques**. Master's thesis, Pontifical Catholic University of Rio de Janeiro, Brazil, 2018.

4 ACHANCCARAY, P. M.. **Crop Recognition in Tropical Regions based on spatio-temporal Conditional Random Fields from multi-temporal and multi-resolution sequences of remote sensing images**. PhD thesis, Pontifical Catholic University of Rio de Janeiro, 2019.

5 RONNEBERGER, O.; FISCHER, P. ; BROX, T.. **U-net: Convolutional networks for biomedical image segmentation**. In: INTERNATIONAL CONFERENCE ON MEDICAL IMAGE COMPUTING AND COMPUTER-ASSISTED INTERVENTION, p. 234–241. Springer, 2015.

6 JÉGOU, S.; DROZDZAL, M.; VAZQUEZ, D.; ROMERO, A. ; BENGIO, Y.. **The one hundred layers tiramisu: Fully convolutional densenets for semantic segmentation**. In: PROCEEDINGS OF THE IEEE CONFERENCE ON COMPUTER VISION AND PATTERN RECOGNITION WORKSHOPS, p. 11–19, 2017.

7 CHEN, L.-C.; PAPANDREOU, G.; SCHROFF, F. ; ADAM, H.. **Rethinking atrous convolution for semantic image segmentation**. arXiv preprint arXiv:1706.05587, 2017.

8 UNITED NATIONS. **World Population Prospects: The 2017 Revision, Key Findings and Advance Tables.** Working Paper No. ESA/P/WP/248., 2017.

9 LEITE, P.; FEITOSA, R.; FORMAGGIO, A.; DA COSTA, G.; PAKZAD, K. ; SANCHES, I.. **Hidden markov models for crop recognition in remote sensing image sequences.** Pattern Recognition Letters, 32(1):19–26, 2011.

10 THENKABAIL, P.. **Land resources monitoring, modeling, and mapping with remote sensing.** CRC Press, 2015.

11 SANCHES, I.; FEITOSA, R.; ACHANCCARAY, P.; SOARES, M.; LUIZ, A.; SCHULTZ, B. ; MAURANO, L.. **Campo verde database: Seeking to improve agricultural remote sensing of tropical areas.** IEEE Geoscience and Remote Sensing Letters, 15(3):369–373, 2018.

12 LECUN, Y.; BENGIO, Y. ; HINTON, G.. **Deep learning.** Nature, 521(7553):436, 2015.

13 AUDEBERT, N.; BOULCH, A.; RANDRIANARIVO, H.; LE SAUX, B.; FERECATU, M.; LEFÈVRE, S. ; MARLET, R.. **Deep learning for urban remote sensing.** In: 2017 JOINT URBAN REMOTE SENSING EVENT (JURSE), p. 1–4. IEEE, 2017.

14 LA ROSA, L. E. C.; HAPP, P. N. ; FEITOSA, R. Q.. **Dense fully convolutional networks for crop recognition from multitemporal sar image sequences.** In: IGARSS 2018-2018 IEEE INTERNATIONAL GEOSCIENCE AND REMOTE SENSING SYMPOSIUM, p. 7460–7463. IEEE, 2018.

15 NDIKUMANA, E.; HO TONG MINH, D.; BAGHDADI, N.; COURAULT, D. ; HOSSARD, L.. **Deep recurrent neural network for agricultural classification using multitemporal sar sentinel-1 for camargue, france.** Remote Sensing, 10(8), 2018.

16 CASTRO, J. B.; FEITOSA, R. Q. ; HAPP, P. N.. **An hybrid recurrent convolutional neural network for crop type recognition based on multitemporal sar image sequences.** In: IGARSS 2018-2018 IEEE INTERNATIONAL GEOSCIENCE AND REMOTE SENSING SYMPOSIUM, p. 3824–3827. IEEE, 2018.

17 XINGJIAN, S.; CHEN, Z.; WANG, H.; YEUNG, D.; WONG, W. ; WOO, W.. **Convolutional lstm network: A machine learning approach for precipitation nowcasting.** In: ADVANCES IN NEURAL INFORMATION PROCESSING SYSTEMS, p. 802–810, 2015.

18 RUSSWURM, M.; KÖRNER, M.. **Multi-temporal land cover classification with sequential recurrent encoders.** ISPRS International Journal of Geo-Information, 7(4):129, 2018.

19 MELGANI, F.; BRUZZONE, L.. **Classification of hyperspectral remote sensing images with support vector machines.** IEEE Transactions on geoscience and remote sensing, 42(8):1778–1790, 2004.

20 PAL, M.. **Random forest classifier for remote sensing classification.** International Journal of Remote Sensing, 26(1):217–222, 2005.

21 NITZE, I.; SCHULTHESS, U. ; ASCHE, H.. **Comparison of machine learning algorithms random forest, artificial neural network and support vector machine to maximum likelihood for supervised crop type classification.** Proc. of the 4th GEOBIA, 35, 2012.

22 JIN, Y.; LIU, X.; CHEN, Y. ; LIANG, X.. **Land-cover mapping using random forest classification and incorporating ndvi time-series and texture: a case study of central shandong.** International Journal of Remote Sensing, 39(23):8703–8723, 2018.

23 HAO, P.; WANG, L. ; NIU, Z.. **Comparison of hybrid classifiers for crop classification using normalized difference vegetation index time series: A case study for major crops in north xinjiang, china.** PloS one, 10(9):e0137748, 2015.

24 USTUNER, M.; SANLI, F.; ABDIKAN, S.; ESETLILI, M. ; KURUCU, Y.. **Crop type classification using vegetation indices of rapideye imagery.** The International Archives of Photogrammetry, Remote Sensing and Spatial Information Sciences, 40(7):195, 2014.

25 USTUNER, M.; BALIK SANLI, F.. **Polarimetric target decompositions and light gradient boosting machine for crop classification: A comparative evaluation.** ISPRS International Journal of Geo-Information, 8(2):97, 2019.

26 BLASCHKE, T.. **Object based image analysis for remote sensing.** ISPRS journal of photogrammetry and remote sensing, 65(1):2–16, 2010.

27 DURO, D. C.; FRANKLIN, S. E. ; DUBÉ, M. G.. **A comparison of pixel-based and object-based image analysis with selected machine learning algorithms for the classification of agricultural landscapes using spot-5 hrg imagery.** Remote sensing of environment, 118:259–272, 2012.

28 LONG, J. A.; LAWRENCE, R. L.; GREENWOOD, M. C.; MARSHALL, L. ; MILLER, P. R.. **Object-oriented crop classification using multitemporal etm+ slc-off imagery and random forest**. GIScience & Remote Sensing, 50(4):418–436, 2013.

29 HAGENSIEKER, R.; ROSCHER, R.; ROSENTRETER, J.; JAKIMOW, B. ; WASKE, B.. **Tropical land use land cover mapping in pará (brazil) using discriminative markov random fields and multi-temporal terrasar-x data**. International journal of applied earth observation and geoinformation, 63:244–256, 2017.

30 ACHANCCARAY, P.; FEITOSA, R. Q.; ROTTENSTEINER, F.; SANCHES, I. ; HEIPKE, C.. **Spatial-temporal conditional random field based model for crop recognition in tropical regions**. In: 2017 IEEE INTERNATIONAL GEOSCIENCE AND REMOTE SENSING SYMPOSIUM (IGARSS), p. 3007–3010. IEEE, 2017.

31 GU, Y.; WANG, Y. ; LI, Y.. **A survey on deep learning-driven remote sensing image scene understanding: Scene classification, scene retrieval and scene-guided object detection**. Applied Sciences, 9(10):2110, 2019.

32 MAKANTASIS, K.; KARANTZALOS, K.; DOULAMIS, A. ; DOULAMIS, N.. **Deep supervised learning for hyperspectral data classification through convolutional neural networks**. In: 2015 IEEE INTERNATIONAL GEOSCIENCE AND REMOTE SENSING SYMPOSIUM (IGARSS), p. 4959–4962. IEEE, 2015.

33 KUSSUL, N.; LAVRENIUK, M.; SKAKUN, S. ; SHELESTOV, A.. **Deep learning classification of land cover and crop types using remote sensing data**. IEEE Geoscience and Remote Sensing Letters, 14(5):778–782, 2017.

34 LONG, J.; SHELHAMER, E. ; DARRELL, T.. **Fully convolutional networks for semantic segmentation**. In: PROCEEDINGS OF THE IEEE CONFERENCE ON COMPUTER VISION AND PATTERN RECOGNITION, p. 3431–3440, 2015.

35 WEI, S.; ZHANG, H.; WANG, C.; WANG, Y. ; XU, L.. **Multi-temporal sar data large-scale crop mapping based on u-net model**. Remote Sensing, 11(1):68, 2019.

36 MARMANIS, D.; WEGNER, J. D.; GALLIANI, S.; SCHINDLER, K.; DATCU, M. ; STILLA, U.. **Semantic segmentation of aerial images with an ensemble of cnns**. ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences, 3:473, 2016.

37 HOCHREITER, S.; SCHMIDHUBER, J.. **Long short-term memory**. Neural computation, 9(8):1735–1780, 1997.

38 CHO, K.; VAN MERRIËNBOER, B.; GULCEHRE, C.; BAHDANAU, D.; BOUGARES, F.; SCHWENK, H. ; BENGIO, Y.. **Learning phrase representations using rnn encoder-decoder for statistical machine translation**. arXiv preprint arXiv:1406.1078, 2014.

39 BERMUDEZ, J.; FEITOSA, R.; ACHANCCARAY, P.; HAPP, P.; SANCHES, I. ; CUÉ, L.. **Evaluation of recurrent neural networks for crop recognition from multitemporal remote sensing images**. In: ANAIS DO XXVII CONGRESSO BRASILEIRO DE CARTOGRAFIA, 2017.

40 RUSSWURM, M.; KÖRNER, M.. **Multi-temporal land cover classification with long short-term memory neural networks**. The International Archives of Photogrammetry, Remote Sensing and Spatial Information Sciences, 42:551, 2017.

41 LILLESAND, T.; KIEFER, R. W. ; CHIPMAN, J.. **Remote sensing and image interpretation**. John Wiley & Sons, 2015.

42 MCNAUGHT, A. D.. **Compendium of chemical terminology**, volumen 1669.

43 MCNAIRN, H.; BRISCO, B.. **The application of c-band polarimetric sar for agriculture: A review**. Canadian Journal of Remote Sensing, 30(3):525–542, 2004.

44 STINCHCOMBE, M.; WHITE, H.. **Universal approximation using feedforward networks with non-sigmoid hidden layer activation functions**. In: IJCNN INTERNATIONAL JOINT CONFERENCE ON NEURAL NETWORKS, 1989.

45 IOFFE, S.; SZEGEDY, C.. **Batch normalization: Accelerating deep network training by reducing internal covariate shift**. arXiv preprint arXiv:1502.03167, 2015.

46 SRIVASTAVA, N.; HINTON, G.; KRIZHEVSKY, A.; SUTSKEVER, I. ; SALAKHUTDINOV, R.. **Dropout: a simple way to prevent neural**

networks from overfitting. The journal of machine learning research, 15(1):1929–1958, 2014.

47 GOODFELLOW, I.; BENGIO, Y. ; COURVILLE, A.. **Deep learning**. MIT press, 2016.

48 NIELSEN, M. A.. **Neural networks and deep learning**, volumen 25. Determination press San Francisco, CA, USA:, 2015.

49 OLIVEIRA, G. L.; BURGARD, W. ; BROX, T.. **Efficient deep models for monocular road segmentation**. In: 2016 IEEE/RSJ INTERNATIONAL CONFERENCE ON INTELLIGENT ROBOTS AND SYSTEMS (IROS), p. 4885–4891. IEEE, 2016.

50 XIE, W.; NOBLE, J. A. ; ZISSERMAN, A.. **Microscopy cell counting and detection with fully convolutional regression networks**. Computer methods in biomechanics and biomedical engineering: Imaging & Visualization, 6(3):283–292, 2018.

51 CHEN, W.; FU, Z.; YANG, D. ; DENG, J.. **Single-image depth perception in the wild**. In: ADVANCES IN NEURAL INFORMATION PROCESSING SYSTEMS, p. 730–738, 2016.

52 HUANG, G.; LIU, Z.; VAN DER MAATEN, L. ; WEINBERGER, K. Q.. **Densely connected convolutional networks**. In: PROCEEDINGS OF THE IEEE CONFERENCE ON COMPUTER VISION AND PATTERN RECOGNITION, p. 4700–4708, 2017.

53 ZHU, Y.; NEWSAM, S.. **Densenet for dense flow**. In: 2017 IEEE INTERNATIONAL CONFERENCE ON IMAGE PROCESSING (ICIP), p. 790–794. IEEE, 2017.

54 KHENED, M.; ALEX, V. ; KRISHNAMURTHI, G.. **Densely connected fully convolutional network for short-axis cardiac cine mr image segmentation and heart diagnosis using random forest**. In: INTERNATIONAL WORKSHOP ON STATISTICAL ATLASES AND COMPUTATIONAL MODELS OF THE HEART, p. 140–151. Springer, 2017.

55 CHEN, L.-C.; PAPANDREOU, G.; KOKKINOS, I.; MURPHY, K. ; YUILLE, A. L.. **Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs**. IEEE transactions on pattern analysis and machine intelligence, 40(4):834–848, 2017.

56 CHEN, L.-C.; ZHU, Y.; PAPANDREOU, G.; SCHROFF, F. ; ADAM, H.. **Encoder-decoder with atrous separable convolution for semantic image segmentation.** In: PROCEEDINGS OF THE EUROPEAN CONFERENCE ON COMPUTER VISION (ECCV), p. 801–818, 2018.

57 LIN, M.; CHEN, Q. ; YAN, S.. **Network in network.** arXiv preprint arXiv:1312.4400, 2013.

58 YANG, M.; YU, K.; ZHANG, C.; LI, Z. ; YANG, K.. **Denseaspp for semantic segmentation in street scenes.** In: PROCEEDINGS OF THE IEEE CONFERENCE ON COMPUTER VISION AND PATTERN RECOGNITION, p. 3684–3692, 2018.

59 PARDO, E.; MORGADO, J. M. T. ; MALPICA, N.. **Semantic segmentation of mfish images using convolutional networks.** Cytometry Part A, 93(6):620–627, 2018.

60 MA, C.; CHEN, M.; KIRA, Z. ; ALREGIB, G.. **Ts-lstm and temporal-inception: Exploiting spatiotemporal dynamics for activity recognition.** Signal Processing: Image Communication, 71:76–87, 2019.

61 SCHUSTER, M.; PALIWAL, K.. **Bidirectional recurrent neural networks.** IEEE Transactions on Signal Processing, 45(11):2673–2681, 1997.

62 SANCHES, I.; FEITOSA, R.; ACHANCCARAY, P.; MONTIBELLER, B.; LUIZ, A.; SOARES, M.; PRUDENTE, V.; VIEIRA, D. ; MAURANO, L.. **Lem benchmark database for tropical agricultural remote sensing application.** ISPRS-International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences, 621:387–392, 2018.

63 SHEN, Y.; WU, L.; DI, L.; YU, G.; TANG, H.; YU, G. ; SHAO, Y.. **Hidden markov models for real-time estimation of corn progress stages using modis and meteorological data.** Remote Sensing, 5(4):1734–1753, 2013.

64 MASSRUHA, S.; OTHERS. **Uma nova abordagem para diagnóstico de doenças via web.** In: EMBRAPA INFORMÁTICA AGROPECUÁRIA-ARTIGO EM ANAIS DE CONGRESSO (ALICE). In: CONGRESSO BRASILEIRO DE AGROINFORMÁTICA, 8., 2011, Bento Gonçalves . . . , 2011.