Pontifícia Universidade Católica
do Rio de Janeiro

Laura Elena Cué La Rosa

End-to-end Convolutional Neural Network combined with Conditional Random Fields for Crop Mapping from Multitemporal SAR Imagery

Tese de Doutorado

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

Advisor : Prof. Raul Queiroz Feitosa
Co-advisor: Prof. Dario Augusto Borges Oliveira

Rio de Janeiro
September 2022

Pontifícia Universidade Católica
do Rio de Janeiro

Laura Elena Cué La Rosa

End-to-end Convolutional Neural Network combined with Conditional Random Fields for Crop Mapping from Multitemporal SAR Imagery

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

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

Prof. Dario Augusto Borges Oliveira
Co-advisor
GE Centro Brasileiro de Pesquisas –

Prof. Ieda Del'Arco Sanches
Instituto Nacional de Pesquisas Espaciais – INPE

Prof. Jefersson A. dos Santos
Universidade Federal de Minas Gerais – UFMG

Prof. Wesley Nunes Gonçalves
Universidade Federal de Mato Grosso do Sul – UFMS

Prof. Matheus Pinheiro Ferreira
Instituto Militar de Engenharia – IME

Rio de Janeiro, September the 26th, 2022

Laura Elena Cué La Rosa

Received her bachelor degree in Biomedical Engineering at Higher Polytechnic Institute "Jose Antonio Echeverria", Havana, Cuba in 2013. She obtained her master's degree in Electrical Engineering with emphasis on Signal Processing and Control at the Pontifícia Universidade Católica do Rio de Janeiro (PUC-Rio) in 2018. Her professional interests comprise deep learning methods (DL) applied to Remote Sensing (RS) image analysis with focus in agriculture and forest mapping.

Bibliographic data

Cue La Rosa, L. E.

End-to-end Convolutional Neural Network combined with Conditional Random Fields for Crop Mapping from Multitemporal SAR Imagery / Laura Elena Cué La Rosa ; advisor: Raul Queiroz Feitosa ; co-advisor: Dario Augusto Borges Oliveira. – 2022.

96 f. : il. color. ; 30 cm

Tese (doutorado)–Pontifícia Universidade Católica do Rio de Janeiro, Departamento de Engenharia Elétrica.

Inclui bibliografia

1. Engenharia Elétrica – Teses. 2. Reconhecimento de Culturas;. 3. Sensoriamento Remoto;. 4. Aprendizado Profundo;. 5. Modelos Graficos Probabilisticos;. 6. Sentinel-1;. I. Feitosa, R. Q.. II. Borges Oliveira, D. A.. III. Pontifícia Universidade Católica do Rio de Janeiro. Departamento de Engenharia Elétrica. IV. Título.

CDD: 620.11

Acknowledgments

My utmost appreciation to my advisor, Prof. Raul Queiroz Feitosa, for his generous support, his advice and leadership throughout the development of this thesis.

I am truly grateful to my co-advisor Prof. Dário Augusto Borges Oliveira for his support during my internship at IBM Research and my journey as a doctoral student. For his encouragement, advice, and stimulating talks.

I thank the members of staff at Computer Vision Lab at PUC-Rio for sharing their company, friendship and valuable scientific advices.

I thank PUC-Rio and CNPq for the financial support.

I want to thank my family and friends from their love and support throughout my life, and most importantly, my appreciation to my mother, Carmen Elena who have always encouraged me and to whom I dedicate this work.

This study was financed in part by the Coordenação de Aperfeiçoamento de Pessoal de Nível Superior - Brasil (CAPES) - Finance Code 001.

Abstract

Cue La Rosa, L. E.; Feitosa, R. Q. (Advisor); Borges Oliveira, D. A. (Co-Advisor). **End-to-end Convolutional Neural Network combined with Conditional Random Fields for Crop Mapping from Multitemporal SAR Imagery**. Rio de Janeiro, 2022. 96p. Tese de doutorado – Departamento de Engenharia Elétrica, Pontifícia Universidade Católica do Rio de Janeiro.

Remote sensing imagery enables accurate crop mapping and monitoring, supporting efficient and sustainable agricultural practices to ensure food security. However, accurate crop type identification and crop area estimation from remote sensing data in tropical regions are still challenging tasks. Compared to the characteristic conditions of temperate regions, the more favorable weather conditions in tropical regions permit higher flexibility in land use, planning, and management, which implies complex crop dynamics. Moreover, the frequent cloud cover prevents the use of optical data during large periods of the year, making SAR data an attractive alternative for crop mapping in tropical regions. To exploit both spatial and temporal contex, conditional random fields (CRFs) models have been used successfully in the classification of RS imagery. These approaches deliver high accuracies; however, they rely on features engineering manually designed based on domain-specific knowledge. In this context, deep learning methods such as convolutional neural networks (CNNs) proved to be a robust alternative for remote sensing image classification, as they can learn optimal features and classification parameters directly from raw data. This work introduces a novel end-to-end hybrid model based on deep learning and conditional random fields for crop recognition in areas characterized by complex spatio-temporal dynamics typical of tropical regions. The proposed framework consists of two modules: a CNN that models spatial and temporal contexts from the input data and a CRF that models temporal dynamics considering label dependencies between adjacent epochs. These dependencies can be learned or designed by an expert in local agricultural practices. Comparisons between data-driven and prior-knowledge temporal constraints are presented for two municipalities in Brazil, using multi-temporal SAR image sequences. The experiments showed significant improvements in per class F1 score of up to 30% and up to 12% in average F1 score against a baseline model that doesn't include temporal dependencies during the learning process.

Keywords

Crop Recognition; Remote Sensing; Deep Learning; Probabilistic Graphical Models; Sentinel-1;

Resumo

Cue La Rosa, L. E.; Feitosa, R. Q.; Borges Oliveira, D. A.. **Treinamento ponta a ponta de redes neurais convolucionais combinadas com campos aleatórios condicionais para o mapeamento de culturas a partir de imagens SAR multitemporais**. Rio de Janeiro, 2022. 96p. Tese de Doutorado – Departamento de Engenharia Elétrica, Pontifícia Universidade Católica do Rio de Janeiro.

Imagens de sensoriamento remoto permitem o monitoramento e mapeamento de culturas de maneira precisa, apoiando práticas de agricultura eficientes e sustentáveis com o objetivo de garantir a segurança alimentar. No entanto, a identificação do tipo de cultura a partir de dados de sensoriamento remoto em regiões tropicais ainda são consideradas tarefas com alto grau de dificuldade. As favoráveis condições climáticas permitem o uso, planejamento e o manejo da terra com maior flexibilidade, o que implica em culturas com dinâmicas mais complexas. Além disso, a presença constante de nuvens dificulta o uso de imagens ópticas, tornando as imagens de radar uma alternativa interessante para o mapeamento de culturas em regiões tropicais. Os modelos de campos aleatórios condicionais (CRFs) têm sido usados satisfatoriamente para explorar o contexto temporal e espacial na classificação de imagens de sensoriamento remoto. Estes modelos oferecem uma alta precisão na classificação, no entanto, dependem de atributos extraídos manualmente com base em conhecimento especializado do domínio. Neste contexto, os métodos de aprendizado profundo, tais como as redes neurais convolucionais (CNNs), provaram ser uma alternativa robusta para a classificação de imagens de sensoriamento, pois podem aprender atributos ótimos diretamente dos dados. Este trabalho apresenta um modelo híbrido baseado em aprendizado profundo e CRF para o reconhecimento de culturas em áreas de regiões tropicais caracterizadas por ter uma dinâmica espaço–temporal complexa. O framework proposto consiste em dois módulos: uma CNNs que modela o contexto espacial e temporal dos dados de entrada, e o CRF que modela a dinâmica temporal considerando a dependência entre rótulos para datas adjacentes. Estas dependências podem ser aprendidas ou desenhadas por um especialista nas práticas de agricultura local. Comparações entre diferentes variantes de como modelar as transições temporais são apresentadas usando sequências de imagens SAR de duas municipalidades no Brasil. Os experimentos mostraram melhorias significativas atingindo ate 30% no F1 score por classe e ate 12% no F1 score medio em relação ao modelo de base que não inclui dependências temporais durante o processo de aprendizagem.

Palavras-chave

Reconhecimento de Culturas; Sensoriamento Remoto; Aprendizado Profundo; Modelos Graficos Probabilisticos; Sentinel-1;

Table of contents

List of figures

List of tables

1
Introduction

1.1
Motivation

Recent reports on food security estimate that over 800 million people worldwide can be considered malnourished and that approximately two billion suffer from deficiencies in micronutrients [6]. Furthermore, with the expected increase in human population from 7.7 billion in 2019 to 8.5 billion in 2030 [7], coupled with the predicted worldwide growth of per capita income, the demand for food is expected to escalate in the near future [8, 9].

The consequent intensification of agricultural production to meet such high projected demands may, however, have strong environmental impacts [8], contributing directly to air and water pollution, soil degradation, and greenhouse gas (GHG) emission, as well as biodiversity loss [9]. In fact, land conversion from natural ecosystems to agriculture is one of the principal causes of GHG emissions and is directly related to deforestation and biodiversity loss [10]. Therefore, there is an urgent need to conceive efficient and sustainable strategies for the agricultural sector that maximize crop productivity while minimizing environmental impact to enhance food security for the current and future human population. Thus, timely and accurate information about cropping practices is crucial to achieving this goal.

Accurate estimation of crop area extents and crop type distribution, for instance, is indispensable for irrigation management, yield prediction, mapping soil productivity, and farm monitoring. Climate and weather unsurprisingly affect cropping areas and, hence, agricultural practices. In temperate regions, agriculture is strongly characterized by seasonality, and the analysis of crop dynamics is simplified by the fact that there is usually a single crop per parcel during the productive season. Crop dynamics in tropical regions are considerately more complex, as multiple harvests per year are possible, and due to particular practices such as crop rotation, non-tillage, and irrigation [4].

Crop area estimation must be provided periodically from the beginning of the growing season until harvest. Identifying large-scale agricultural areas

is possible due to the improved spatial and temporal resolution of remote sensing data associated with the increased computational capacity and the advancements in classification methods.

Remote sensing (RS) data has been used in natural resources mapping for many decades, being currently the main data source for various environmental modeling techniques, which include crop recognition and crop area estimation [11–18]. Notwithstanding, automatic crop mapping is still a complex problem, especially in tropical regions. An important issue is related to the fact that the spectral appearance of crops changes over time. Consequently, multi-temporal analysis is crucial for crop discrimination, especially in regions characterized by complex and diverse crop dynamics.

Optical and synthetic aperture radar (SAR) orbital systems with high temporal (low revisit time) and spatial resolutions are a valuable asset for crop mapping applications. While optical imagery commonly supports agricultural applications [19, 20], they are often unavailable during the growing season due to cloud cover, especially in tropical and subtropical regions [21]. In contrast, SAR sensors are not affected by clouds or solar illumination conditions and therefore available most of the year, making them an attractive option for multi-temporal analysis for crop mapping applications.

Traditional classification techniques for RS images generally make use of unsupervised (e.g., k-means) or supervised (e.g., maximum likelihood, neural network, support vector machine, random forest) methods to perform pixel-wise classification [22–27]. However, these approaches lack of a proper framework that explicitly models the spatial and temporal context. Spatio-contextual techniques such as texture extraction and object-based classification have been widely used for RS classification [28–34]. Nevertheless, the discriminative ability of these models strongly depends on the choice of features to be used in the classification procedure.

To cope with the inherent problems of pixel-wise and object-based approaches, probabilistic graphical models, such as Markov random fields, hidden Markov models (HMMs), and conditional random fields (CRFs), have successfully exploited both spatial and temporal contexts in the classification of RS imagery [35–39]. These approaches deliver high accuracies but they rely on feature engineering, and we argue that there are no universal hand-crafted features, equally discriminative for different applications and datasets. In graphical models, the temporal context is modeled using a transition matrix that can be learned directly from the training data [40], or based on expert-based knowledge [38, 39, 41]. The definition of the transition matrix must follow the agricultural practices and crop rotations characteristics of the target sites.

Deep learning (DL) techniques encompass specific supervised and unsupervised representation-learning algorithms, which learn features from labeled and non-labeled data. State-of-the-art performance in RS image classification has been achieved with DL-based techniques, such as autoencoders (AEs), convolutional neural networks (CNNs), fully convolutional networks (FCNs), and recurrent neural networks (RNNs); which can integrate the spatial, spectral and temporal contexts in unsupervised or supervised ways [42–47].

Specifically, a CNN [48, 49] is a neural network capable of dealing with spatial and temporal context and has been used with great success in RS. In an early work, Kussul et al. [44] integrated spatial and temporal contexts in a supervised way using a combination of 1D and 2D CNNs. Castro et al. [50] also used AEs and CNNs for crop recognition in multi-temporal SAR image sequences. Considering both the temporal and spatial context, 3D CNN has also been successfully employed for crop mapping from multi-temporal remote sensing images [51, 52].

Several efforts have been made in the literature to combine graphical and deep learning models into a two-stage procedure for crop type classification [5, 53, 54]. However, despite achieving competitive performance, these works have the drawback of decoupling the classifier training from the graphical model. These deep learning classifiers are commonly trained using the per-date categorical cross-entropy loss function. Therefore, the classification at each time step is conditionally independent of the class on the neighbor's dates.

To learn a model specifically for sequence labeling, CNNs or RNNs have been combined with graphical models (e.g., CRF) in an end-to-end fashion to model the intra-class temporal progression. These models were first proposed for sequence tagging for natural language processing and then extended for video action segmentation [55–57]. These approaches are trained using a CRF-based loss that considers the probability of a particular label sequence given an input sequence. To this aim, the models employ a CNN that delivers the emission scores of the CRF model and a learnable transition score matrix that models the temporal relation between adjacent labels.

Building on that, this work proposes a novel hybrid method that combines deep learning and graphical models in an end-to-end framework for crop mapping in tropical regions from multi-temporal SAR image sequences. To achieve this goal, CNN and CRF were employed to model the spatio-temporal context. Furthermore, considering the complex temporal dynamic characteristic of tropical regions, this study proposes using prior knowledge about the less probable crop transitions to add temporal constraint into the learning process. It is worth mentioning that the methodology is not limited to SAR data,

the method could be easily applied to other type of sensors as long as enough temporal data is available.

1.2 Problem statement

Crop mapping using multi-temporal RS images is based on a discrete acquisition of data from a continuous process consisting of sequences of phenological cycles from various crops planted in a given order. However, most current DL methods learn features disregarding the explicit modeling of such sequences in the temporal dimension and implement the knowledge-based constraints as a post-processing step that filters the outcome to select the valid ones. We argue that such feature space learned is suboptimal as it ignores the known conditional class dependency between neighboring dates in the learning process, leading to further suboptimal classification models for crop mapping.

1.3 Research hypotheses

First, we want to analyze if combining deep learning and graphical models in an end-to-end fashion would increase the final classification performance. Also, we want to explore the model behavior under different temporal constraints, one that directly learns the inter-class relationship from the data and the other that incorporates prior knowledge about crop dynamics into the learning process. Thus, the following research hypotheses are defined:

1. Including a CRF-based loss function into the learning process of a CNN model could potentially improve the per-date classification.

2. Learning the transition scores conditioned to the training data could be restrictive when different label sequences are observed on the test set. Hence, adding the temporal constraint based on prior knowledge about the possible and less probable crop transitions could improve model generalization to unseen label sequences.

3. Implementing temporal constraints for less probable classes could potentially lead to more efficient models that use prior knowledge support for less obvious information from the training data.

4. Combining losses for full sequence and individual dates could improve per-date classification compared to models trained solely with one of them.

1.4 Objectives

General Objective The general objective of this work is to propose an end-to-end convolutional neural network that incorporates prior knowledge about local crop temporal dynamics for more efficient crop recognition models in tropical regions using sequences of SAR images.

Specific Objectives In pursuit of the general objective and the research hypotheses, the specific objectives are:

1. Propose a solution for embedding prior knowledge about local crop dynamics in an end-to-end CNN training schema using conditional random fields.

2. Exploit both spatial and temporal contexts using a hybrid end-to-end deep learning model based on CNN and CRF.

3. Evaluate the model capabilities and limitations for learning complex crop dynamics in tropical areas from learning data.

4. Evaluate the benefits of adding temporal constraints based on explicit prior knowledge for improving model generalization.

5. Evaluate the impact of combining full sequence and per-date losses to improve the performance of the final per-date classification model.

1.5 Contributions

The main contributions of this work are the following:

1. A novel end-to-end hybrid method based on CNN and CRF for crop mapping in tropical regions from sequences of remote sensing images.

2. A model setup that enables learning a transition matrix with explicit crop dynamics knowledge directly from the training data.

3. A model setup that enables embedding explicit prior knowledge about crop dynamics for improving crop classification models' performance.

1.6 Organization of the remaining parts

The thesis is structured in six chapters. Chapter 2 introduces the reader to basic concepts of SAR imagery and a literature review of crop mapping using graphical models and deep learning methods. Chapter 3 provides the fundamental concepts and theory for to better understand the proposed hybrid method. Chapter 4 introduces and explains the proposed hybrid method for crop recognition based on convolutional neural networks and conditional random fields trained end-to-end. Chapter 5 presents the datasets employed in this work, the experimental protocol, and the results obtained with the proposed hybrid model. Finally, Chapter 6 summarizes the conclusions derived from the performed experiments and provides directions for further development of the proposed method.

2
Crop Mapping from Remote Sensing data

This chapter provides an overview of crop mapping from remote sensing data, including relevant works based on graphical models and deep learning. In addition, a brief introduction to synthetic aperture radar (SAR) and its principles is given.

2.1
Synthetic Aperture Radar

Synthetic aperture radar (SAR) has been widely used for Earth remote sensing for more than 30 years. It provides high-resolution images independent from daylight [58, 59]. As an active RS system, SAR systems transmit radio waves and register the echoes (backscatter) reflected by Earth's surface objects, penetrating vegetation canopy and dry soil. Moreover, the characteristic wavelength ranges of SAR imaging systems enable the transmitted signals to penetrate clouds, making such systems almost insensitive to adverse atmospheric conditions [60], and thus highly reliable in terms of data provisioning [61].

Principles of SAR: SAR systems have a side-looking imaging geometry and are based on a pulsed radar (see Figure 2.1). The system transmits electromagnetic pulses and receives the echoes of the backscattered signal, where its amplitude and phase depend on the image's physical and electrical properties. The system stores the backscatter information corresponding to the cell area on the ground scene. The resulting radar imagery is built up from the strength and time delay of the returned signal. SAR systems use the movement of the radar in orbit to synthesize a virtual long antenna from the short physical antenna in the direction of flight.

The radar signals are either transmitted with the electric field plane parallel (horizontal polarization) or perpendicular (vertical polarization) to the Earth's surface. In this way, the antenna can transmit and receive in either horizontal (H) or vertical (V) single polarization (HH or VV, where the first letter indicates transmit and the second receive) or cross-polarization (HV or VH) [58].

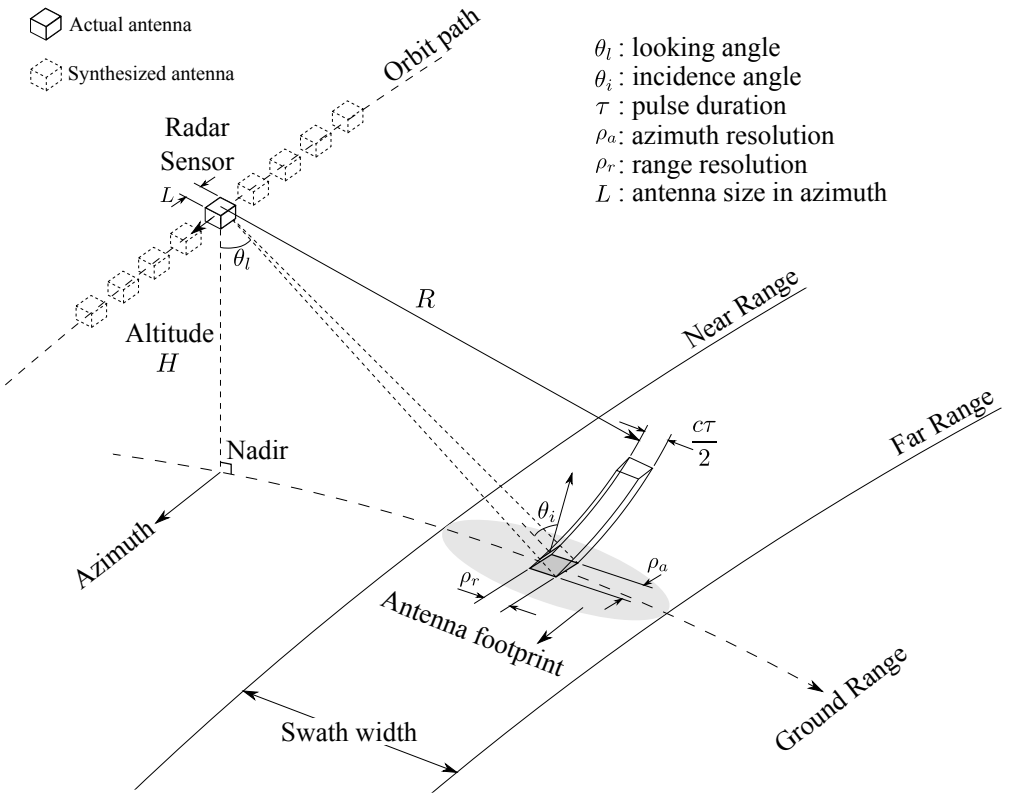

Figure 2.1: Components of a Synthetic Aperture Radar (SAR) system. Source [1].

The images are recorded parallel to the sensor motion (azimuth) and orthogonal to its motion (range) using a side-looking geometry with an oblique angle θ_l and an incidence angle θ_i (see Figure 2.1). The swath width gives the ground-range resolution (ρ_r) of the radar scene that depends on the radar pulse duration τ; and the antenna size provides the azimuth resolution (ρ_a) L [62]. Both resolutions are computed as follows:

$$\rho_r = \frac{c\tau}{2\sin\theta_i}, \quad \rho_a = \frac{L}{2}, \tag{2-1}$$

where c is the speed of light.

These resolutions compose the so-called resolution cell, which is the minimum possible distance between two objects to be distinguished and determines the quality of ground maps generated by the SAR system. The backscatter measured from a target area in SAR is usually normalized per unit geometric cell area known as normalized backscatter coefficient σ^0 as shown in the following equation:

$$\sigma^0 = P_r \frac{(4\pi)^3 * R^3}{A * P_t * G^2 * \lambda^2}. \tag{2-2}$$

where P_r and P_t refer to the received and transmitted energy, respectively, G is the antenna gain, λ is the wavelength, R corresponds to the range from the

antenna to the target and A is the area over which the measurement is made.

SAR images for crop analysis: The backscatter intensities recorded by SAR systems are a function of the size, shape, orientation, and dielectric constant of the scatters [63]. In crop analysis studies, backscatter intensities would therefore differ depending on the particular characteristics of the crop components (leaves, stalks, seeds, etc.) and soil moisture content. Crops with different intrinsic structures can therefore be distinguished up to some point, based on their backscatter intensities [64]. The development of multipolarized acquisition modes in many available systems further increases the discriminative capacity of SAR data [65]. VV polarized signals interact more with crop structure while HH polarization penetrates crops and captures underlying soil roughness and moisture content [66–68]. Usually VV–VH is the preferred dual-polarization for crop classification. Figure 2.2 presents an example of a SAR image with this type of polarization from an agricultural region in Campo Verde, Mato Grosso, Brazil. The image was acquired by Sentinel-1A C-Band SAR, with a spatial resolution of 10 m.

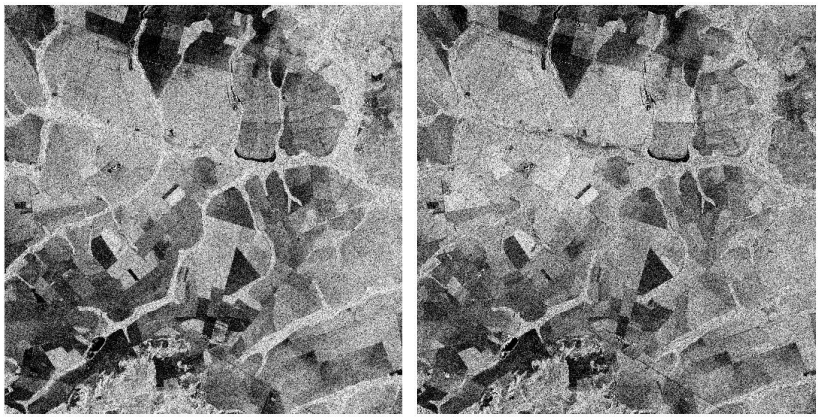

Figure 2.2: Image of Campo Verde agricultural region in Mato Grosso municipality, Brazil. Image acquired with Sentinel-1A C-Band SAR, 10 m spatial resolution, VH and VV bands dual-polarized.

In this work, however, no prior knowledge about the particular characteristics of the SAR image data nor the particular interactions of radar waves with different crop types were used in the definition of the investigated deep learning classification methods, as they are supposed to automatically learn features to be used in the crop classification task.

2.2 Spatio-temporal crop dynamics

Remote sensing satellites enable the acquisition of large-scale images capable of capturing spectral, spatial, and temporal phenomena in agricultural regions. However, crop type classification is challenging due to the significant changes that crops experience during the growing season, both spatially and temporally. Crop type mapping requires insights into the phenological states, seasonal growth, local agricultural practices, and weather conditions. An important issue is related to the fact that the spectral appearance of crops changes over time. For example, different crops may be in the same phenological state, presenting similar spectra, but as the growing season progress, they may differ significantly [37]. Considering specifically SAR data, crops in the same phenological stage present similar SAR backscatter. Consequently, multi-temporal analysis is crucial for crop discrimination, especially in regions characterized by complex and diverse crop dynamics.

Crop rotations, which is a specific sequence of crops in successive periods that can last months to years, is a common practice used by farmers to improve soil conditions and boost system productivity while reducing the demand for fertilizers and pesticides [69, 70]. When applied regularly, this agricultural practice delivers specific temporal patterns that allow predicting the crop type in a given field at a particular time if the previous crop sequence is known.

In temperate regions, the crop dynamics are comparatively simple because there is usually just a single crop per parcel during the whole season, and crop rotations generally occur between successive years. On the other hand, crop dynamics in tropical areas are more complex due to multiple agricultural practices such as irrigation, non-tillage, double cropping systems, crop-livestock rotation, and multiple harvests per year [4, 53]. Crop rotation is common practice in agricultural systems in tropical and subtropical regions. Figure 2.3 presents an example of double cropping with crop rotation maize followed by soybean in São Paulo state, Brazil [2]. More complex crop rotations, for example, triple cropping system, can also be observed for some of the regions [2]. In addition, for specific crop types there is period of sanitary break which consists of 2–3 month period where the crops are absent from the fields in order to prevent rust infection [71]. These agricultural practices make crop type classification from remote sensing data in tropical regions more challenging than in temperate areas.

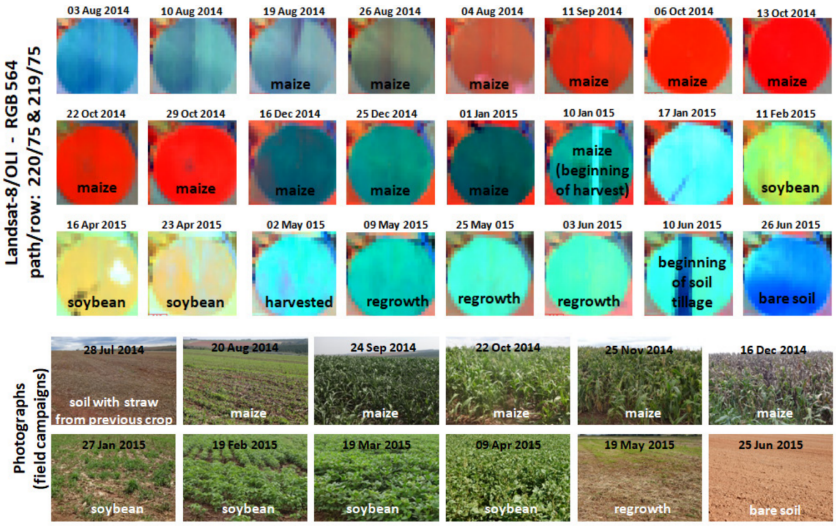

Figure 2.3: Photographs from field campaign and satellite data of an area cultivated with crop rotation of maize followed by soybean. Source [2].

2.3 Graphical Models for crop type mapping

As the spectral appearance representing the same area changes over time, the temporal context is the relationship of an image site (pixel or segment) with respect to different acquisition times. Therefore, incorporating temporal context in the classification model allows for significant improvements in the classification of crops and vegetation [16]. Probabilistic graphical models have been successfully used in the past decade for crop type classification due to their ability to exploit spatio-temporal context from remote sensing images. Early works employed hidden Markov models (HMMs) [72] to learn spectral response variations in time among crop phenology stages [37, 73]. However, HMM lacks a proper way of modeling the spatial context and therefore represents a drawback when used for crop type classification. In contrast, Markov random fields (MRFs) and conditional random fields (CRFs) can explicitly model the spatio-temporal context. In [69] a Markov logic of crop rotations model was proposed for early crop mapping based on machine learning applied to historical data. The framework enables both learning from data and integrating expert knowledge. Hagensieker et al. [74] introduced a spatio-temporal MRF to discriminate among burnt pasture, clean pasture, shrubby pasture, water, and forest. In their formulation, the association potential is delivered by an import vector machines (IVM) classifier, a Potts model gives the spatial interaction potential, and temporal potential is represented by transition matrices from expert knowledge, respectively. In [40], the authors proposed CRFmulti, a multi-temporal CRF that integrates spatial and temporal context for crop type classification from optical images. This study modeled the temporal

context using a global transition matrix learned from training data. In [39], the authors employed expert-based phenology knowledge in a higher-order dynamic CRF for crop type mapping in a temperate region. A straightforward example of expert knowledge that can be integrated into the learning process on a graphical model is the possible rotations (i.e., transition matrices) of the crops in a particular agricultural region. The definition of these site-specific transition matrices must reflect the agricultural landscape and the type of crop rotations, requiring an understanding of the internal dynamics of crops. Castellazzi et al. [41] use Markov chains with transition probabilities set by experts. In [38], a spatio-temporal CRF model was proposed for crop mapping in tropical regions. Here the association potential was given by a random forest classifier, the spatial interaction by a Potts model, and the temporal interaction potential by a transition matrix based on expert knowledge.

2.4 Deep Learning for crop type mapping

Current state-of-the-art performance in crop type mapping from RS image has been achieved with DL-based techniques [5, 42–47]. Such models can be roughly grouped into two categories: convolutional neural networks (CNN) that explore the spatial context and recurrent neural networks (RNN) and Selfattention (transformers) to model multi-temporal data sequences. One of the first works that employed DL to discriminate among different crop types from RS images was presented in [44]. Here the authors proposed a 1D and 2D CNN architectures to exploit spectral and spatial features, respectively. The authors integrated spatial and temporal contexts in a supervised way and concluded that an ensemble of 1D and 2D CNNs outperformed the Random Forest (RF) classifier to discriminate among wheat, maize, soybeans, sugar beet, and sunflower. In [50], the authors also used AEs and CNNs for crop recognition in multi-temporal SAR image sequences, obtaining results that outperformed the RF classifier. [75] combines CNN and LSTM for crop mapping in Brazil using multi-temporal image sequences from Sentinel-1. In [76], the authors explore the performances of 1D CNNs, and to types of RNN (LSTM and GRU) for early crop classification from Sentinel-1A imagery and reported the best Kappa coefficient (0.942) for the 1D CNNs classifier. In [77], the authors compared five deep learning models and found that 1D CNN, LSTM-CNN, and GRU-CNN achieved high accuracy (above 85%) in classifying crop types in China from Sentinel-2 time series images. Attention-based models such as AtLSTM, AtBiLSTM, and Transformer have also been employed with success for crop type mapping from multi-spectral, and multi-

temporal RS imagery [78, 79]. A recent study employed a 2D CNN for crop classification from stacked Landsat-8 images in a tropical region from Brazil [80]. A dual attention convolutional neural network was also proposed for crop classification using time-series Sentinel-2 imagery, achieving a Kappa coefficient of 0.98 outperforming other state-of-the-art classification methods, including RF, XGBOOST, R-CNN, 2D-CNN, and 3D-CNN [81]. Considering 3D CNN, some studies have utilized 3D convolutions for learning spatio-temporal features [82]. Unlike the aforementioned 2D CNN architecture, where temporal information is exploited by stacking the multi-temporal data, the 3D CNN architecture uses 3D kernels to perform 3D convolution operations that can extract spatial and spectral/temporal features simultaneously. In [82], the authors proposed a 3D CNN-based method to classify crops from multi-temporal RS images automatically. The study concluded that 3D CNN is especially suitable for characterizing crop growth dynamics and outperformed the 2D CNN method.

In the aforementioned DL-based approaches, however, the trained network computes a descriptor for either a pixel or a given image patch/parcel and predicts a single label for the entire patch; this label is then assumed to be the label of the patch's central pixel. During the test phase, the map is constructed by predicting each patch associated with each image position. However, this approach can be highly inefficient for large-scale images. Additionally, such an approach is inappropriate for pixel-wise semantic labeling tasks, as it assigns a label to a patch independently of the surrounding labels. This patch-level classification often leads to a salt-and-pepper-like result and limits the power of the network to learn intra- and inter-class spatial relations.

More efficient approaches jointly predict all labels in an image patch instead of a single label for the central pixel. In this scenario, the so-called fully convolutional network (FCNs) came into play. FCNs [83] were specifically proposed for semantic labeling; those networks employ an upsampling strategy at the second half set of layers of a CNN to recover the original input image size and perform dense predictions. The network performs end-to-end learning, downsampling the input space (typically by successive convolution, activation, and pooling layers) and then upsampling (deconvolution) it again to predict dense output labels for an arbitrary size input. In FCNs, learning and inference are performed for the whole image at once to get a probability map of semantic labels without loss in terms of spatial resolution. Since every label is learned in association with its neighbors, the method can be seen as a structured one. Such networks have delivered impressive performances in RS applications, as reported in [84, 85]. In [86], a FCN-based approach was compared with a

CNN-based approach for crop classification in SAR images. The study reports similar thematic and spatial accuracy results for both approaches. In terms of computational cost, however, the inference time of the FCN-based approach was more than one hundred times shorter than that of the CNN approach. In [45], a fully convolutional LSTM (ConvLSTM) was proposed to exploit spatial and temporal information for multi-temporal crop recognition from Sentinel-2 image sequences. Recent studies have employed different versions of FCN such as DenseFCN, UNet, and DeepLabv3+ for crop type classification using multi-temporal SAR images sequence [53, 87, 88]. Although a good performance is reported, these approaches were carried out in a temperate region, where a single crop type occurs in the agricultural year, or requires training an independent FCN for each date/time period for its application in tropical regions where crop dynamics is considerately more complex.

3D FCN approach can successfully combine temporal and spatial context by replacing all the 2D convolutional with 3D convolutions as the building blocks of the FCN, allowing its use to deliver a multi-temporal labeled image sequence. In this context, in [89], a 3D-UNet was implemented for crop type mapping in Africa using SAR and optical time-series images to deliver a classification map for the agricultural year. A 3d FCN was proposed in [90] to discriminate between soybean and corn, obtaining a Kappa coefficient of 91.8%. Recently, [52] proposed a 3D UNet that employs atrous temporal pyramid pooling (ATPP) [91] for multi-date crop type semantic segmentation in a tropical region, outperforming a U-Net with bidirectional ConvLSTM.

The studies mentioned above depend on a large number of densely annotated training samples (i.e., all pixel labels within the input image are known) to deliver an accurate semantic segmentation map, which may restrict their application for large-scale image classification. In the last few years, some alternatives arose to train FCNs with weak supervision, enabling low-cost annotations using points, scribbles, and polygons [92–95]. In [95], the authors proposed to train a FCN from scribbles using a partial cross-entropy (pCE) loss. The proposed loss only back-propagates gradients for the scribble annotated pixels, emerging as an effective approach to deal with low-cost annotations. Considering crop type classification, [88] employed the same loss function to train a 2D FCN for single-date classifications using the stacked multi-temporal sequence of Sentinel-1 images.

2.5 Deep Learning and Graphical Models

Several efforts have been made in the literature to combine graphical models and deep learning. For example, in [53], the authors used a dense FCN for multi-date crop recognition upon a SAR multi-temporal image sequence in a tropical region. They trained separate models for each desired output date and then applied a post-processing algorithm based on a HMM, the so-called most-likely class sequence (MLCS), that enforces prior knowledge about crop occurrence over time in the target region. Martinez et al. [5] proposed a more computationally efficient approach that employs a bidirectional convolutional RNN that considers the spatio-temporal context and delivers a prediction for each date of interest. The authors also applied the MLCS algorithm that further improved the per-date accuracy. Recently, [54] proposed a spatial-temporal HMM for land cover classification that employs a stacked AE that delivers the class probability input to the HMM model. Despite achieving competitive performance, these works have the drawback of decoupling the classifier training from the graphical model. Commonly, these CNN classifiers are trained using the per-date categorical cross-entropy loss function. Therefore, the classification at each time-step is conditionally independent of the class on the neighbors' dates.

3
Theoretical foundations

This chapter presents the theoretical foundations for understanding the frameworks proposed in Chapter 4. First, general concepts of graphical models are explained. Then, the fundamentals of linear-chain Conditional Random Fields (CRFs) are carefully detailed. Finally, some concepts related to Convolutional Neural Networks (CNNs) are described, with the main focus on temporal convolutions. For further details, the reader is referred to the papers cited in the following sections.

3.1
Graphical Models

Graphical models are commonly used to encode a joint or conditional probability distribution between a target random variable we wish to predict and some input random variables that we assume are observed [3, 96]. In this sense, a graphical model represents a family of probability distributions that satisfy all conditional independences encoded in a graphical structure. Graphical models represent the distribution over all random variables by a product of local functions, each depending on a subset of a small number of variables. There are two well-known graphical models, the *directed graphical models* (also known as Bayesian networks) that express causality among variables, and the *undirected graphical models* (also known as Markov random fields (MRF)), that express interaction between variables [97]. A graph is composed of nodes connected by links, where each node represents a group of random variables, and each link expresses the probabilistic relationships among them [97]. In a Bayesian network, the links have a directionality; in Markov random fields, the links are undirected.

Bayesian networks: A Bayesian network is based on a *directed graph* $G = (V, E)$, in which the vertices V denote the set of random variables $\mathbf{x}$ and E the edges that determine the conditional dependence among them. This graph describes the probability distribution over a set $\mathbf{x} = \{x_i\}$ of random variables as a joint distribution in the factorized form

$$p(\mathbf{x}) = \prod_i p(x_i|\mathrm{PA}_i) \tag{3-1}$$

where PA_i is the set of parents of node x_i in G. Here, each node is conditionally independent of all its non-descendants in the graph given the values of all its parents [98, 99]. Therefore this graph is defined by the product of conditional distributions and is always correctly normalized [97]. This is a type of generative model that captures the causal process by which the observed data was generated [3, 97].

Markov random fields: A Markov random field is an *undirected graph* $G = (V, F, E)$ in which V denotes the random variables, F denotes the factors (also called *potential functions*), and E the edges. This graph expresses the joint distribution $p(\mathbf{x})$ as a product of *feature functions* over the maximal cliques of the graph[1] with the following factorization:

$$p(\mathbf{x}) = \frac{1}{Z} \prod_{c \in C} \psi_c(\mathbf{x}_c) \tag{3-2}$$

where C denotes the set of all maximal cliques, ψ_c is the *potential function*, and Z is the partition function (a normalization constant) given by $Z = \sum_{\mathbf{x}} \prod_{c \in C} \psi_c(\mathbf{x}_c)$ that ensures the distribution sums to 1. This normalization constant involves an increase in the computational cost that is exponential to the model's size. Generally, because each *potential function* is strictly positive, is often represented as exponential:

$$\psi_c(\mathbf{x}_c) = \exp\{-E(\mathbf{x}_c)\}, \tag{3-3}$$

where $E(\mathbf{x}_c)$ is the energy function. In contrast to a *directed graph*, these potentials are not restricted to a probabilistic interpretation, which gives more flexibility to the model.

Application for classification: For a classification problem, we want to construct a model that predicts a label y given an input feature vector $\mathbf{x} = \{x_i\}$, assuming that once the label is known all features are independent [3]. The naive Bayes classifier for this problem takes the form:

$$p(y, \mathbf{x}) = p(y)p(\mathbf{x}|y) = p(y) \prod_i p(\mathbf{x}_i|y) \tag{3-4}$$

where $p(\mathbf{x}_i|y)$ is the likelihood of observation $\mathbf{x}_i$ for label y. This model is illustrated by the *directed graphical model* in Figure 3.1 (top left).

In an *undirected graphical model*, the posterior takes the form of a conditional distribution that factorizes as follows:

$$p(y|\mathbf{x}) = \frac{1}{Z(\mathbf{x})} \prod_{c \in C} \psi_c(\mathbf{x}_c, y_c) \tag{3-5}$$

[1] A clique is a set of fully connected nodes, a maximal clique is a clique such that it is not possible to include any other nodes without ceasing to be a clique [97]

where y_c is the set of labels for clique c and $Z(\mathbf{x}) = \sum_{\mathbf{y}'} \prod_{c \in C} \psi_c(\mathbf{x}_c, y'_c)$ is the partition function. This is a discriminative approach.

In the generative approach, we model $p(\mathbf{x}|y)$ and use it together with the Bayes rule for classification. In contrast, in a discriminative approach, we model $p(y|\mathbf{x})$ directly, resulting in a simpler inference problem than modeling the joint distribution. Generative models describe how the output y can generate the input $\mathbf{x}$, whereas discriminative models describe how to assign an output y to a given input $\mathbf{x}$ [3].

A well-known classifier that can be represented by an *undirected graph* is the logistic regression (LR). LR is a discriminative classifier since it is based on a conditional distribution and in standard form

$$p(y|\mathbf{x}) = \frac{1}{Z(\mathbf{x})} \exp\left\{ \sum_i \theta_i f_i(y, \mathbf{x}) \right\}, \tag{3-6}$$

where $Z(\mathbf{x})$ is the normalization constant, and θ_i and f_i are the weights and feature function. Figure 3.1 (bottom left) describes the graph for this classifier. We introduce this notation since it is closely related to the usual notation for conditional random fields.

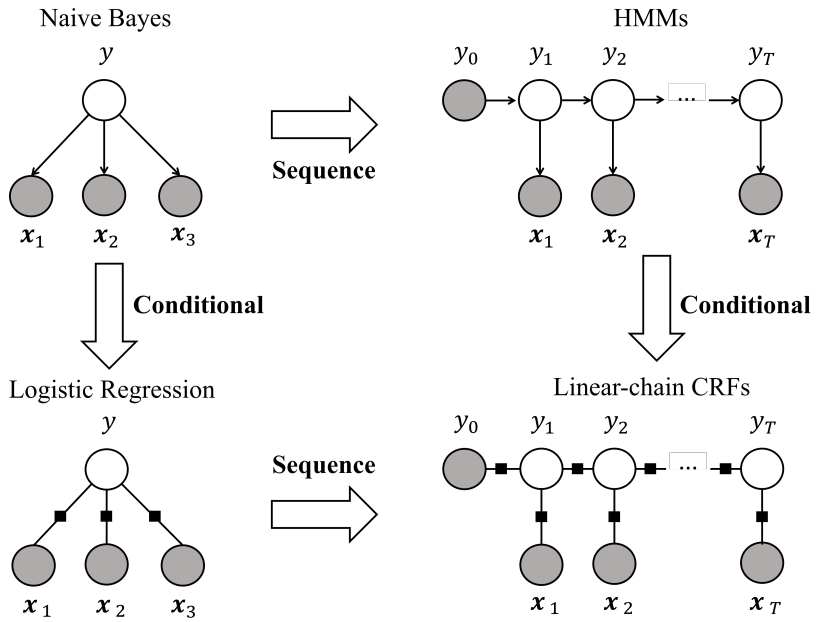

Figure 3.1: Diagram of the relationship between naive Bayes, logistic regression, HMMs, and linear-chain CRFs. The circles are variable nodes, and the black boxes are factor nodes. Adapted from [3].

3.2 Hidden Markov Models

So far, we have described how to use graphical models for classification. However, many real-world applications require structured predictions where we want to predict a structure rather than a unique label. For example, a sequence of labels where neighboring labels are dependent. By ignoring this sequential nature, we can lose a lot of information. Here is when graphical models play an essential role, modeling the variables' dependencies in which the output labels are arranged in a sequence. Time series analysis is one of the applications that benefit the most from these models, including modeling crop dynamics [36] which is the main application of this thesis.

To consider the sequential aspect, first, the independence assumption is relaxed by considering that the output labels are arranged in a linear chain [3] (also known as Markov model). A first-order Markov chain assumes that each conditional distribution on the temporal axis depends only on the most recent observation. Hence, the joint distribution for a sequence of observations $\mathbf{x} = \{x_t\}_{t=1}^{T}$ is given by

$$p(x_1, ..., x_T) = p(x_1) \prod_{t=2}^{T} p(x_t|x_{t-1}), \tag{3-7}$$

and the conditional distribution for observation x_t is given by:

$$p(x_t|x_1, ..., x_{t-1}) = p(x_t|x_{t-1}). \tag{3-8}$$

However, this formulation is still very restrictive, losing valuable information from earlier observations. A solution could be extending the model to a higher order, which implies high computational cost because the parameters grow exponentially with the order. Here is when a hidden Markov model (HMM) [72] comes to play.

A hidden Markov model (HMM) is a type of *directed graphical model* that considers that a given process evolves through a sequence of hidden states $\mathbf{y} = \{y_t\}_{t=1}^{T}$ emitting the observable signals $\mathbf{x} = \{x_t\}_{t=1}^{T}$ where each state still holds the conditional independence property $(y_t \perp\!\!\!\perp y_{t-2})|y_{t-1}$, i.e. each state y_t depends only on its immediate predecessor. That way, there is always a path connecting two observed variables x_t and x_m via the hidden states, which leads to the independence assumption that each observation x_t depends only on the current state y_t. With these assumptions, the join distribution in a HMM writes as

$$p(x_1, ..., x_T, y_1, ..., y_T) = p(y_1) \prod_{t=2}^{T} p(y_t|y_{t-1}) \prod_{t=1}^{T} p(x_t|y_t), \tag{3-9}$$

where $p(x_t|y_t)$ is the likelihood of x_t given the state y_t (also know as emission probabilities) and $p(y_t|y_{t-1})$ is the state transition score from time step $t-1$ to t (transition probabilities), and $p(y_1)$ denotes the initial probability distribution over sates. Without loss of generality, we can write the initial prior probability $p(y_1)$ as $p(y_1|y_0)$, and rewrite the HMM as:

$$p(\mathbf{x}, \mathbf{y}) = \prod_{t=1}^{T} p(y_t|y_{t-1})p(x_t|y_t), \tag{3-10}$$

Figure 3.1 (top right) describes the graph for the above formulation HMMs. One shortcoming of a HMM is that being a generative model it requires learning the conditional probability distributions $p(x_t|y_t)$ while in a sequence labeling task, we are interested in the conditional probability $p(\mathbf{y}|\mathbf{x})$. Moreover, the HMM model only captures dependencies between each state and its corresponding observation.

3.3 Linear-Chain Conditional Random Field

A linear-chain conditional random field (CRF) combines discriminative and sequence modeling by considering a conditional distribution that follows the joint distribution of a HMM [3]. In this sense, a linear-chain CRF is like a HMM but with a set of factors (i.e., the *feature functions*) that are not necessarily probability distributions.

Lets denote the input sequence of observations as $\mathbf{x} = \{x_t\}_{t=1}^{T}$, the output sequence as $\mathbf{y} = \{y_t\}_{t=1}^{T} : \mathbf{y} \in \mathcal{Y}$, where $\mathcal{Y}$ is the set of all generic label sequences, and the set of possible labels as S. The CRF way of defining the factors over this set of random variables is taking an exponential of a linear combination of feature functions f with parameters $\theta = \theta_1 \cup \theta_2$:

$$\begin{aligned} p(\mathbf{y}|\mathbf{x}) &= \frac{1}{Z(\mathbf{x})} \prod_{t=1}^{T} \exp\{\theta_1 f_{em}(y_t, \mathbf{x}_t)\} \exp\{(\theta_2 f_{tr}(y_t, y_{t-1})\} \\ &= \frac{1}{Z(\mathbf{x})} \exp\left\{ \sum_{t=1}^{T} \theta_1 f_{em}(y_t, \mathbf{x}_t) + \sum_{t=1}^{T} \theta_2 f_{tr}(y_t, y_{t-1}) \right\}, \end{aligned} \tag{3-11}$$

where f_{em} and f_{tr} are the feature functions to obtain the emission and transition scores, respectively, θ_1 and θ_2 are the weights, and $Z(\mathbf{x})$ is the partition function that ensures the normalization between 0 and 1. Here, the partition function is formulated as:

$$Z(\mathbf{x}) = \sum_{\mathbf{y}'} \exp\left\{ \sum_{t=1}^{T} \theta_1 f_{em}(y_t', \mathbf{x}_t) + \sum_{t=1}^{T} \theta_2 f_{tr}(y_t', y_{t-1}') \right\}. \tag{3-12}$$

The feature functions f_{em} and f_{tr} are typically defined based on knowledge about the application. In speech tagging task for example, a feature func-

tion $f_{tr}(y_t, y_{t-1})$ could take values of 1 if y_{t-1} is an adjective and y_t is a noun; and 0 otherwise. A positive weight θ_2 for this feature means that adjectives tend to be followed by nouns. In computer vision tasks, specifically in remote sensing image analysis, the feature function f_{rm} could be defined as texture features extracted from the GLCM matrix. Moreover, these feature functions can also be learned automatically from the input data using deep learning methods.

General definition: The CRF defined in Equation 3-11 is similar to the LR classifier in Equation 3-6, and its factor graph is illustrated in Figure 3.1 (bottom right). Note that, as $\mathbf{x}$ is observed, we can condition on it and take a slice of the emission factors, which leads to the standard definition of a linear-chain CRF

$$p(\mathbf{y}|\mathbf{x}) = \frac{1}{Z(\mathbf{x})} \exp\left\{ \sum_{t=1}^{T} \theta_1 f_{em}(y_t, \mathbf{x}, t) + \sum_{t=1}^{T} \theta_2 f_{tr}(y_t, y_{t-1}) \right\}, \tag{3-13}$$

where the partition function writes as

$$Z(\mathbf{x}) = \sum_{\mathbf{y}'} \exp\left\{ \sum_{t=1}^{T} \theta_1 f_{em}(y_t^{'}, \mathbf{x}, t) + \sum_{t=1}^{T} \theta_2 f_{tr}(y_t^{'}, y_{t-1}^{'}) \right\} \tag{3-14}$$

The first sum in the exponent of the conditional distribution $p(\mathbf{y}|\mathbf{x})$, represents the score of assigning a particular y_t at time-step t given the input observation $\mathbf{x}$, and the second sum represents the score that each y_t follows a particular state y_{t-1}. In such a way, a linear-chain CRF permits rich, overlapping features of the observed variable $\mathbf{x}$, i.e., the feature functions f_{em} can depend on observations from any time-step [3]. Figure 3.2 illustrates the graphical model for the above formulation.

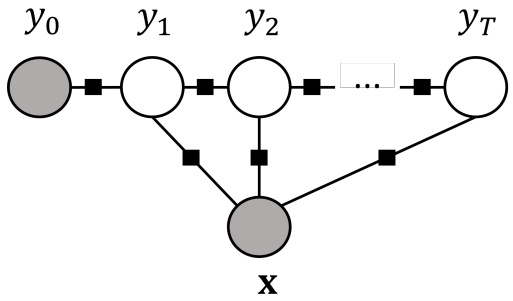

Figure 3.2: Factor graph of a linear-chain CRFs where each feature take a slice of the emission factors that is needed for computing features at time-step t.

3.3.1 Training

Given the standard definition of a linear-chain CRFs in Equation 3-13, the next step is to learn the feature weights $\theta = \theta_1 \cup \theta_2$. One way to train the model is by using the maximum likelihood. Since we are modeling the conditional distribution $p(\mathbf{y}|\mathbf{x})$, an appropriate form to estimate the parameters is using the log-likelihood [100]:

$$\begin{aligned}\mathcal{L}(\theta) &= \log p(\mathbf{y}|\mathbf{x},\theta) \\ &= \log\left[\frac{1}{Z(\mathbf{x})}\exp\left\{\sum_{t=1}^{T}\theta_1 f_{em}(y_t,\mathbf{x},t)+\sum_{t=1}^{T}\theta_2 f_{tr}(y_t,y_{t-1})\right\}\right] \\ &= \log\left[\exp\left\{\sum_{t=1}^{T}\theta_1 f_{em}(y_t,\mathbf{x},t)+\sum_{t=1}^{T}\theta_2 f_{tr}(y_t,y_{t-1})\right\}\right]+\log\left[\frac{1}{Z(\mathbf{x})}\right] \\ &= \left(\sum_{t=1}^{T}\theta_1 f_{em}(y_t,\mathbf{x},t)+\sum_{t=1}^{T}\theta_2 f_{tr}(y_t,y_{t-1})\right)-\log Z(\mathbf{x}).\end{aligned} \tag{3-15}$$

Above, $\mathbf{y}$ is the known true label sequence of the training sample $\mathbf{x}$. Given a set of data samples $\mathbf{A} = \{\mathbf{x}^i, \mathbf{y}^i\}_{i=1}^{A}$, we can use gradient descent to optimize the parameters [3]. The partial derivative of the first term can be interpreted as the expected value of the features under the empirical distribution, and the derivation of the second term is the expectation under the model distribution [3, 100]:

$$\nabla_\theta \mathcal{L}(\theta) = \tilde{E}(f_{em}, f_{tr}) - E(f_{em}, f_{tr}) \tag{3-16}$$

This way, the model learns the parameters that predict the features that are found in the same distribution as in the reference sequences, and when the sequence likelihood is maximized, the gradient tends to zero.

Computing the partition function: An important remark is the computational cost of computing the partition function $Z(\mathbf{x})$ that requires summing over all possible target sequences. Note that, if computing naively, Equation 3-14 implies a time complexity of $O(|S|^T)$. However, it can be computed more efficiently using dynamic programming using belief propagation (BP) [72]. BP is an inference algorithm that generalizes the forward-backward algorithm and can be implemented using sum-product.

To compute $Z(\mathbf{x})$ efficiently, we use the forward algorithm. For brevity, we rewrite $Z(\mathbf{x})$ using factors $\psi_{(.)}$ and apply the distributive law as follow:

$$
\begin{aligned}
Z(\mathbf{x}) &= \sum_{\mathbf{y}'} \left(\prod_{t=1}^{T} \psi_{em}(y_t', \mathbf{x}, t) \psi_{tr}(y_t', y_{t-1}') \right) \\
&= \sum_{y_1'} \sum_{y_2'} \cdots \sum_{y_T'} \psi_{em}(y_1', \mathbf{x}, 1) \psi_{em}(y_2', \mathbf{x}, 2) \cdot \cdot \\
&\quad \cdots \psi_{em}(y_T', \mathbf{x}, T) \psi_{tr}(y_1', y_0') \psi_{tr}(y_2', y_1') \cdots \psi_{tr}(y_T', y_{T-1}')
\end{aligned}
\tag{3-17}
$$

and rearranging the sums and taking each factor as far forward as possible

$$
\begin{aligned}
Z(\mathbf{x}) = &\sum_{y_T'} \psi_{em}(y_T', \mathbf{x}, T) \psi_{tr}(y_T', y_{T-1}') \sum_{y_{T-1}'} \cdots \\
&\cdots \sum_{y_2'} \psi_{em}(y_2', \mathbf{x}, 2) \psi_{tr}(y_2', y_1') \sum_{y_1'} \psi_{em}(y_1', \mathbf{x}, 1) \psi_{tr}(y_1', y_0').
\end{aligned}
\tag{3-18}
$$

The above equation is a sum of products where each intermediate sum is reused many times, and therefore we can store these values for future use. This leads to defining a set of forward functions α's that stores the intermediate sums using the following recursion:

$$
\alpha_t(y_t') = \sum_{y_t'} \psi_{em}(y_t', \mathbf{x}, t) \psi_{tr}(y_t', y_{t-1}') \alpha_{t-1}(y_{t-1}'), \tag{3-19}
$$

where $t = \{1, ..., T\}$ are the time steps, $\alpha_t(y_t')$ is a vector of size $|S|$ where each entry represents a sum over $|S|$ of y_{t-1}', and $\alpha_1(y_1') = \sum_{y_1'} \psi_{em}(y_1', \mathbf{x}, 1) \cdot \psi_{tr}(y_1', y_0')$. Finally, the partition function reads as

$$
Z(\mathbf{x}) = \sum_{y_T'} \alpha_T(y_T'), \tag{3-20}
$$

which gives a final time complexity of $O(T|S|^2)$, much lower than the naive approach. The same principle can be applied to obtain the marginals for the gradients, but this time the summation is in reverse order, defining the backward functions β's as:

$$
\beta_t(y_t') = \sum_{y_{t+1}'} \psi_{em}(y_{t+1}', \mathbf{x}, t+1) \psi_{tr}(y_{t+1}', y_t') \beta_{t+1}(y_{t+1}'), \tag{3-21}
$$

where $t = \{T-1, ..., 1\}$ and $\beta_T(y_T') = 1$. Similarly to the forward pass, the partition function can be computed as

$$
Z(\mathbf{x}) = \beta_0(y_0') = \sum_{y_1'} \psi_{em}(y_1', \mathbf{x}, 1) \psi_{tr}(y_1', y_0'). \tag{3-22}
$$

The forward-backward algorithm is linear in the sequence length and quadratic in the number of labels S; hence the time complexity is $O(T|S|^2)$. The α functions are messages sent from the beginning of the chain to the end, and the β functions are messages sent from the end to the beginning [100]. By combining the forward and backward ways of computing the partition

function it is possible to compute the expectation under the model distribution $E(f_{em}, f_{tr})$ efficiently [101].

3.3.2 Inference

At inference time, the aim is to find the most likely label sequence $\hat{\mathbf{y}}$ given an unseen input sequence $\mathbf{x}$, i.e., the maximum scoring sequence according to our model. For this we must solve the following equation:

$$\hat{\mathbf{y}} = \arg\max_{\mathbf{y}\in\mathcal{Y}} p(\mathbf{y}|\mathbf{x}). \tag{3-23}$$

Solving this equation will require computing the probability for all possible sequences, which is computationally infeasible. Instead, we can use dynamic programming again. The algorithm that solves the above formulation is the Viterbi algorithm [72]. The Viterbi algorithm has the same time complexity as BL, $O(T|S|^2)$. It also consists of a forward-pass, but instead of calculating the sums, the algorithm takes the maximum. This yields the Viterbi recursion [100]:

$$v_t(y_t^{'}) = \max_{y_t^{'}} \psi_{em}(y_t^{'}, \mathbf{x}, t)\psi_{tr}(y_t^{'}, y_{t-1}^{'})v_{t-1}(y_{t-1}^{'}), \tag{3-24}$$

where $v_1(y_1^{'}) = \max_{y_1^{'}} \psi_{em}(y_1^{'}, \mathbf{x}, 1)\psi_{tr}(y_1^{'}, y_{t-0}^{'})$. At each time-step, we keep track of the previous $y^{'}$ that maximized the current recursion and store these values as "back-pointers" to be used in the backward pass:

$$b_t(y_t^{'}) = \arg\max_{y_t^{'}} \psi_{em}(y_t^{'}, \mathbf{x}, t)\psi_{tr}(y_t^{'}, y_{t-1}^{'})b_{t-1}(y_{t-1}^{'}), \tag{3-25}$$

where $b_1(y_1^{'}) = \arg\max_{y_1^{'}} \psi_{em}(y_1^{'}, \mathbf{x}, 1)\psi_{tr}(y_1^{'}, y_0^{'})$. During backward, the Viterbi algorithm decodes starting at the last position in the sequence by finding the label which maximizes the score at the last time-step $\hat{y}_T^{'} = \arg\max_{y_T^{'}} v_T(y_T^{'})$ and follows the backpointers to get the best backward path

$$\hat{y}_t^{'} \leftarrow b_{t+1}(\hat{y}_{t+1}^{'}) \quad \forall t \in \{T-1, ..., 1\}. \tag{3-26}$$

Finally, $\hat{\mathbf{y}} = \{\hat{y}_1^{'}, \hat{y}_2^{'}, ..., \hat{y}_T^{'}\}$. The conditional independence relations allow for such efficient decoding without calculating the likelihood of all possible sequences. Further details about this algorithm can be found in [102].

3.4 Convolutional neural networks

A convolutional neural network (CNN) [48, 49] is a neural network capable of dealing with spatial context. In image analysis, CNNs are typically employed for assigning a single class label to an entire image/scene. The CNN forward pass involves the sequential processing of many layers, thus learning

a hierarchy of feature representations. Its typical building blocks are linear convolution operations followed by non-linear activation, spatial pooling, fully connected layers, and a classification layer.

The main building block of a CNN model is the convolutional layer. A convolutional layer consists of a set of trainable filters applied to local receptive fields (i.e., the regions of the input space that are path-connected to the filter) to extract (interesting) features.

3.4.1 2D convolutional layer

This type of layer consists of a set of K' learnable three-dimensional filters $w \in R^{N \times M \times K}$ (also called kernels), with N and M the size of the horizontal and vertical spatial dimensions and K the depth dimension, respectively; that maps a K dimensional input data to a new K' dimensional activation map. The essential characteristic of the convolutional layer is that all input spatial locations are subjected to the same filters. As each filter is applied by sliding it over the input, the number of parameters to be learned is relatively small compared to traditional neural networks. Given an input data $x \in R^{H \times W \times K}$, where H and W are the height and width, respectively, and K is the input channels; the 2D convolutional responses at the spatial coordinate i, j for the k'-th filter is:

$$x'_{ij,k'} = \sum_{c=0}^{K-1} \sum_{n=0}^{N-1} \sum_{m=0}^{M-1} (w_{n,m,k} * x_{i+n,j+m,k} + b), \tag{3-27}$$

where b is a bias vector. The centers of the convolutional operation are selected using a sliding windows technique with a user-selected stride (s) parameter. Finally, the output feature map of the convolutional layer x' has dimension $(\frac{H-N+2z}{s} + 1, \frac{W-M+2z}{s} + 1, K')$, where z is the zero padding in the spatial dimension. Figure 3.3 (top) represents an example of 2D convolution where the output preserves the same spatial resolution as the input. The small square in the input indicates the spatial portion weighted by the kernels. Each convolution generates one pixel in the output feature map indicated by the colored tiny squares. One way to reduce the spatial dimension is to set the stride larger than one.

3.4.2 3D convolutional layer

If we now consider a spatio-temporal input data $x \in R^{D \times H \times W \times K}$, where D is the temporal dimension, we can follow the same principle to compute features from both spatial and temporal dimensions using a 3D convolution

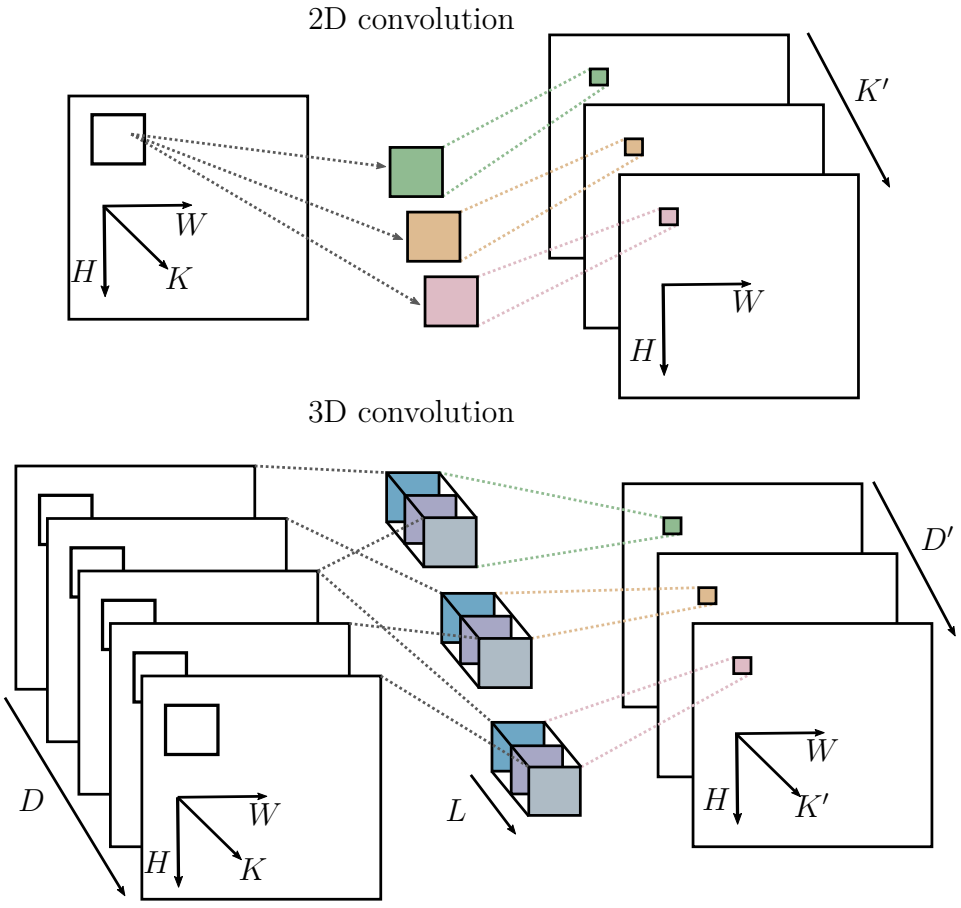

Figure 3.3: Representation for 2D (top) and 3D (bottom) convolutions.

operation. Formally, given a set of K' four-dimensional filters $w \in R^{L \times N \times M \times K}$, with L the size of kernel in the temporal dimension, the activation map for the spatial coordinate e, i, j for the k'-th filter becomes

$$x'_{eij,k'} = \sum_{l=0}^{L-1} \sum_{k=0}^{K-1} \sum_{n=0}^{N-1} \sum_{m=0}^{M-1} (w_{l,n,m,k} * x_{e+l,i+n,j+m,k} + b). \tag{3-28}$$

This operation is applied to each spatial portion of the input data x using a sliding window with a predefined stride for the spatial (s) and temporal (t) dimensions. The output feature maps x' takes the dimension $(\frac{D-L+2p}{t}+1, \frac{H-N+2z}{s}+1, \frac{W-M+2z}{s}+1, K')$, where z and p are the zero padding in the spatial and temporal dimensions, respectively. Figure 3.3 (bottom) represents an example of 3D convolution, where the output preserves the same spatial resolution as the input and $D' = \frac{D-L+2p}{t}+1$. For simplicity, we represent the operation for one kernel; once all the K' kernels process the input data, the output will be D' features maps with dimension $H \times W \times K'$.

3.4.3 Others processing layers

A batch normalization (BN) layer and a non-linear activation function are commonly applied after a convolutional layer. BN [103] forces the set of features throughout a network to have zero mean and unit variance for each training mini-batch. The non-linear activation functions are applied to introduce non-linearity to the process. The most common activation functions are: *sigmoid*, *tanh*, *ReLU* and *Leaky ReLU*. The pooling layer is a downsampling layer. Its objective is twofold: to provide some shift-invariance and to summarize spatial information while preserving discrimination, both at a low computational cost. It consists of mapping each non-overlapping subregion (typically 2×2) to a single number, the maximum or the average within the group. A fully connected layer is commonly used at the end of a CNN model and implies that every neuron in the previous layer is connected to every neuron of the next layer. In sequence comes the classification layer, which delivers scores (class membership probabilities) that are usually determined by the softmax activation function.

3.4.4 Fully Convolutional Networks

Semantic segmentation (or semantic labeling) is the process of classifying each pixel in an image. Currently, state-of-the-art methods for semantic segmentation in remote sensing images are based on FCN architecture [104, 105], usually implementing land-cover usage applications. In FCN [83], typical CNN fully connected layers are replaced by convolutional layers and upsampling operations, avoiding redundant operations in overlapping image tiles and performing structured predictions. These networks consist of an encoder stage that extracts high- and low-level semantic features from convolutions, non-linear activation functions, and downsampling layers; and a decoder stage that uses convolutions, non-linear activation functions, and upsampling layers to produce a target output with the spatial dimension of the input image. The network is trained end-to-end by example without the need for user-specific knowledge. The decoder module allows the exploration of multi-level context information and learns shape and inter- and intra-class variability in the training images. However, blurred boundaries and low spatial resolution that affect the discrimination of object details are common problems [105]. Many strategies have been proposed to tackle those issues, such as skip connections [106], atrous convolutions [107] and pyramid scene parsing pooling [108]. The so-called skip connections transfer local information by concatenating feature maps from

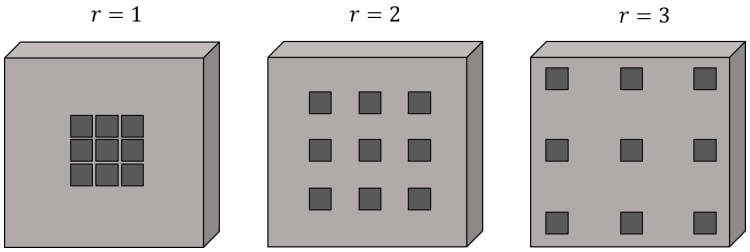

Figure 3.4: Example of atrous convolutions with different atrous rates.

the downsampling path with feature maps from the upsampling path. These connections combine context/semantic information with spatial/appearance information.

The DeepLabv3+ architecture [109], which is considered state-of-the-art for this task, is based on atrous convolutions and pyramid scene parsing pooling. In [109], the authors employed the Atrous Spatial Pyramid Pooling (ASPP) module [107], which consists of parallel atrous convolutions operations. The atrous convolution's fundamental characteristic is the filters with $r - 1$ rows/columns of zeros separating neighboring learnable weights, as shown in Figure 3.4, when r is the atrous rate that determines the minimum distance between two learnable filter weights. With x as the input, the atrous convolution is defined as:

$$y[i] = \sum_{k=1}^{K} x[i + r * k]w[x] \tag{3-29}$$

where $y[i]$ is the output feature map at pixel i, w is the convolutional filter and r is the atrous rate. Notice that when $r = 1$, atrous convolution is equivalent to the standard convolution. This convolution allows a larger receptive field without increasing the number of parameters or loss in spatial resolution. Performing these operations in parallel permits capturing context at multiple scales.

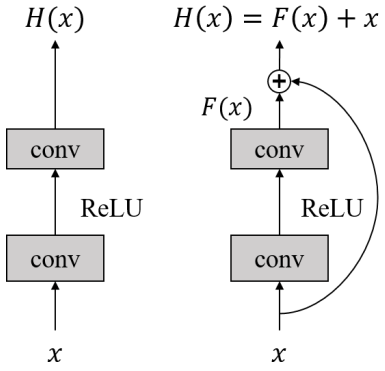

Figure 3.5: A regular block (left) and a residual block (right).

Moreover, residual connections (ResNet) enable training deeper networks

bypassing the vanishing gradients problem [110]. ResNet networks [110] learn not an underlying mapping function $H(x)$ (Figure 3.5 (left)) but a residual function $H(x) - x$ that is expected to be more discriminant. In its final form, the residual block learns a function $H(x) = F(x) + x$ (Figure 3.5 (right)), and uses a shortcut connection that handles the gradient vanishing problem without adding any extra parameter to the network.

4
Proposed hybrid model for crop type mapping

This chapter presents the methodology for making dynamic decisions for multi-date crop recognition using linear-chain CRF and temporal CNN, called *CNN-CRF* hereafter. In this study, we propose several *CNN-CRF* model variants depending on how the temporal dynamic is modeled. The variants can be roughly grouped into two training schemes: data-driven and prior knowledge. For the data-driven variants, we train the framework to learn the transition scores directly from the training data. For the prior knowledge variants, we impose temporal constraints based on expert knowledge about transitions between crops that may or may not occur. We propose these two training schemes to assess the methodology under two real-life situations. The first one considers a scenario when we only have the training data and no information about the crop dynamics in the region is known; the second assumes a scenario when prior knowledge about the less probable crop transitions is available. Finally, we also propose a last training setting that trains the schemes mentioned above in a multi-loss fashion; for this, we employ the CRF-based loss and a per-date cross-entropy loss. In this sense, we extend the studies in [5, 53] and propose a hybrid end-to-end framework. We also implement a partial loss function that allows training the network with scarcely annotated training sets.

Each variant and the baseline model are resumed in Table 4.1. We describe the general framework and give more details about each one of the variants in the following sections.

4.1
General Framework

The proposed end-to-end framework named *CNN-CRF* is presented in Figure 4.1 and consists of three modules: a convolutional neural network (CNN), a linear-chain CRF, and a Viterbi algorithm that delivers the final sequence for each pixel at inference time. The CNN module can be any CNN capable of modeling spatial and temporal context, such as 3D CNN, 3D FCN, or LSTM-CNN networks that have been used with success for crop mapping from multi-temporal remote sensing images [5, 111]. In a recent study,

Table 4.1: Variants of the *CNN-CRF* framework and baseline models.

	Schema	Model name	Characteristic
non-hybrid	Baseline1	*CNN*	per-date cross-entropy loss no label dependency
	Baseline2	*CNN-Vit*	post-processing with Viterbi algorithm consider label dependency
hybrid	Schema 1 data-driven	$CNN\text{-}CRF_L$	learning transitions consider label dependency
	Schema 2 prior-knowledge	$CNN\text{-}CRF_F$	fix possible and impossible transitions consider label dependency
		$CNN\text{-}CRF_P$	fix only impossible transitions consider label dependency
	Schema 3: multi-loss	$MCNN\text{-}CRF_{(.)}$	schema 1 and schema 2 models train with crf and cross-entropy losses

Rogozinski et al. [52] reported superior performance of 3D FCN over LSTM-FCN for crop type classification from multi-temporal SAR image sequences in tropical regions. For this reason, we limit our study to a 3D FCN as the CNN module.

The framework takes as inputs a multi-temporal remote sensing image sequence $\mathbf{x} \in R^{T\times H\times W\times K}$ where each image at each time-step covers the same region and delivers a multi-temporal label image sequence $\mathbf{y} \in R^{T\times H\times W}$. Here, H and W represent the height and width, respectively; and T and K are the length of the sequence and the number of bands, respectively.

Deep learning models can deliver the emission scores directly from the input data. In this study, the image sequence is fed to the CNN module that learns spatio-temporal features and for each pixel produces a score for each label at each time step. We denote the output of the CNN at each time-step as $U(\mathbf{x}, \mathbf{y}_t, t) \in R^{T\times H\times W\times S}$, where S is the number of classes and $\mathbf{y}_t$ is the labels at time-step t. These are the emission scores (also regarded as unary scores) that serve as input to the CRF module. Basically, it is related to the posterior probability of the input sequence at time-step t.

The transition scores are part of the CRF module and are contained in a transition matrix $\mathbf{Tr}$ with size $(T-1\times|S|\times|S|)$, where each $\mathbf{Tr}_t$ is the transition score matrix considering the two adjacent epochs $t-1$ and t; and S is the number of classes. For each $\mathbf{Tr}_t$, elements in row i and column j relates to the probability of the input sequence being the i-th crop type at time-step t, considering the j-th crop type in the previous time-step $t-1$. Considering this, the proposed CRF module can be formulated as

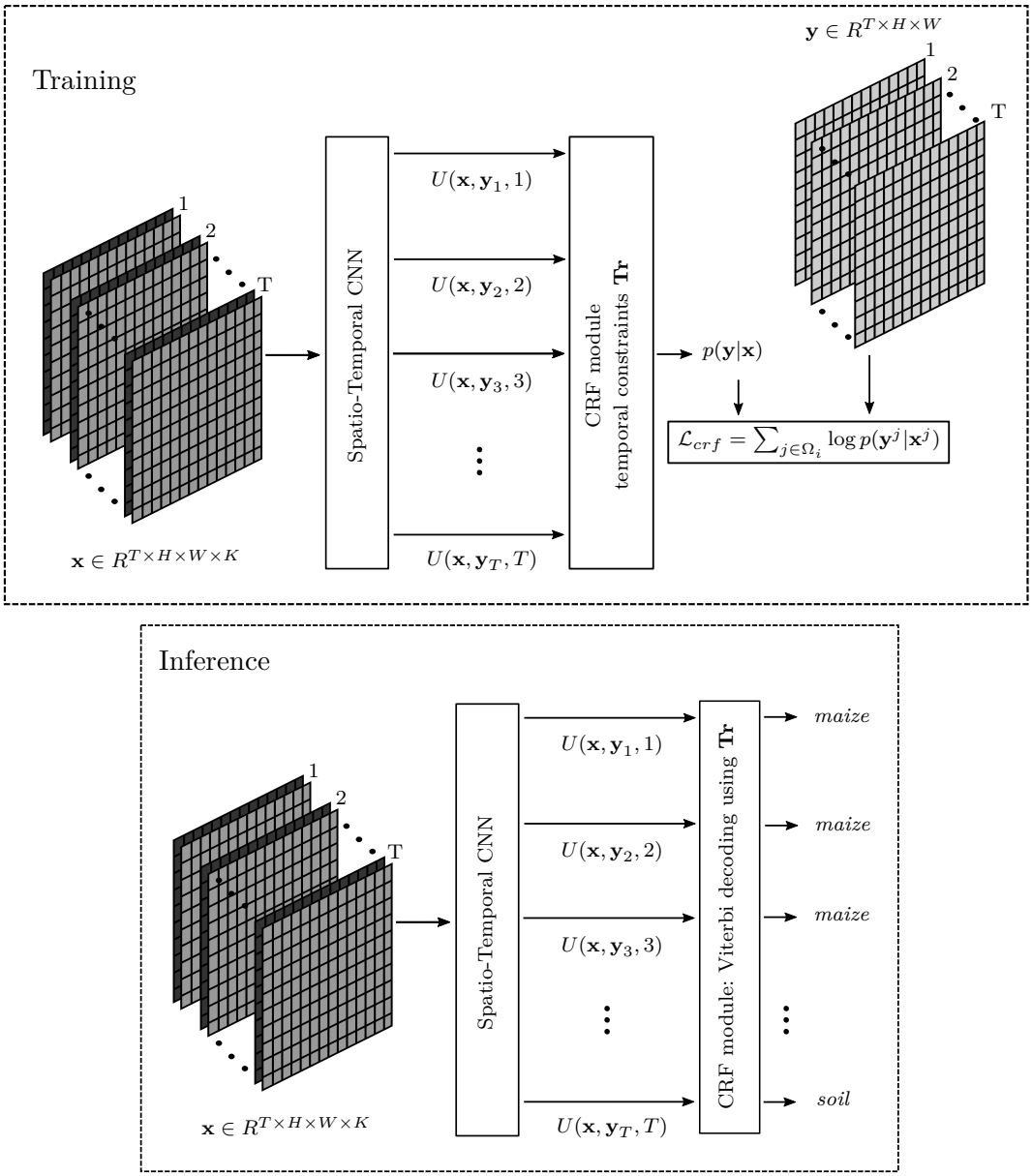

Figure 4.1: General framework of the proposed *CNN-CRF* method. The network is trained end-to-end using the CRF loss function. At inference, a Viterbi algorithm is applied to find the most likely sequence using the emission and transition scores.

$$p(\mathbf{y}|\mathbf{x}) = \frac{1}{Z(\mathbf{x})} \exp\left\{ \sum_{t=1}^{T} U(\mathbf{x}, y_t, t) + \sum_{t=1}^{T} \mathbf{Tr}[t, y_t, y_{t-1}] \right\}, \tag{4-1}$$

where the partition function reads as

$$Z(\mathbf{x}) = \sum_{\mathbf{y}'} \exp\left\{ \sum_{t=1}^{T} U(\mathbf{x}, y_t', t) + \sum_{t=1}^{T} \mathbf{Tr}[t, y_t', y_{t-1}'] \right\}. \tag{4-2}$$

Both the CNN and CRF modules are trained end-to-end by minimizing the negative log-likelihood (NLL), which is equivalent to maximizing the log-likelihood of our data. Given a set of training samples pair $\mathbf{A} = \{\mathbf{x}^i, \mathbf{y}^i\}_{i=1}^{A}$:

$\mathbf{y} \in \mathcal{Y}$, where $\mathcal{Y}$ is the set of all label sequences, the NLL loss function reads as:

$$\begin{aligned}\mathcal{L}_{crf} &= -\sum_{i=1}^{A}\sum_{j\in\Omega_i} \log p(\mathbf{y}^{ij}|\mathbf{x}^{ij}) \\ &= \sum_{i=1}^{A}\sum_{j\in\Omega_i} \log Z(\mathbf{x}^{ij}) \\ &- \sum_{i=1}^{A}\sum_{j\in\Omega_i} \left(\sum_{t=1}^{T} U(\mathbf{x}^{ij}, y_t^{ij}, t) + \sum_{t=1}^{T} \mathbf{Tr}[t, y_t^{ij}, y_{t-1}^{ij}] \right).\end{aligned} \tag{4-3}$$

Here Ω_i, with $|\Omega_i| = WH$, is the set of labeled pixels for training image sample i. At inference we employ the Viterbi decoding to obtain the most likely sequence $\hat{\mathbf{y}} = \arg\max_{\mathbf{y}\in\mathcal{Y}} p(\mathbf{y}|\mathbf{x})$ for each pixel within the input image. For this, we use the emission scores and the transition scores.

4.2 CNN-CRF variants

4.2.1 Data-driven transition scores

The end-to-end CNN-CRF model was first proposed for natural language processing task [55, 56] and assumes that the transition matrix is shared over time, i.e., a global matrix whose transitions scores are independent of the time-step. Despite the success of similar models for sequence tagging in natural language problems, learning a global transition matrix can be too restrictive for crop phenology changes, principally in tropical regions. The disadvantage of using a global matrix for crop type classification was reported in [112]. In this sense, we propose a variant called *CNN-CRF*$_L$, that *learns* a transition matrix for each pair of adjacent epochs (considering a fixed period in the year) conditioned to the observed transitions in the training set.

After training, given an unseen multi-temporal image sequence, *CNN-CRF*$_L$ employs the Viterbi algorithm to obtain the most likely class sequence. For this, the prediction of the CNN and the learned transition matrices are used as emission and transition scores, respectively.

4.2.2 Adding prior knowledge

Training a data-driven model, such as the one mentioned above, will require a large number of annotated samples that comprehend all possible sequences for a given region. However, creating such datasets in tropical areas with complex crop dynamics and different agricultural practices is unfeasible.

The graphs in Figure 4.2 represent crop transitions observed in an hypothetical target site with three classes, *maize*, *soybean*, and *soil*. The Figure presents two transition graphs, one for the training samples (Figure 4.2a) and one for the test samples (Figure 4.2c), considering epochs t and $t+1$. In tropical regions such as Brazil, before rotating from one crop to another, there is a period of harvesting followed by sowing characterized by soil presence. This restriction is expressed in the graphs by the lack of edges corresponding to the transition *maize* $\rightarrow$ *soybean* and *soybean* $\rightarrow$ *maize*. Furthermore, the absence of edges leading to *maize* in $t+1$ indicates that *maize* should never occurs at epoch $t+1$. Trained in such dataset, the *CNN-CRF*$_L$ model will learn a transition matrix (Figure 4.2b) that represents these forbidden transitions (indicated by the negative numbers). At the same time, however, the so-trained model will misclassify any test sequence containing the transition *soybean* $\rightarrow$ *soil*, as it did not occur in the training data.

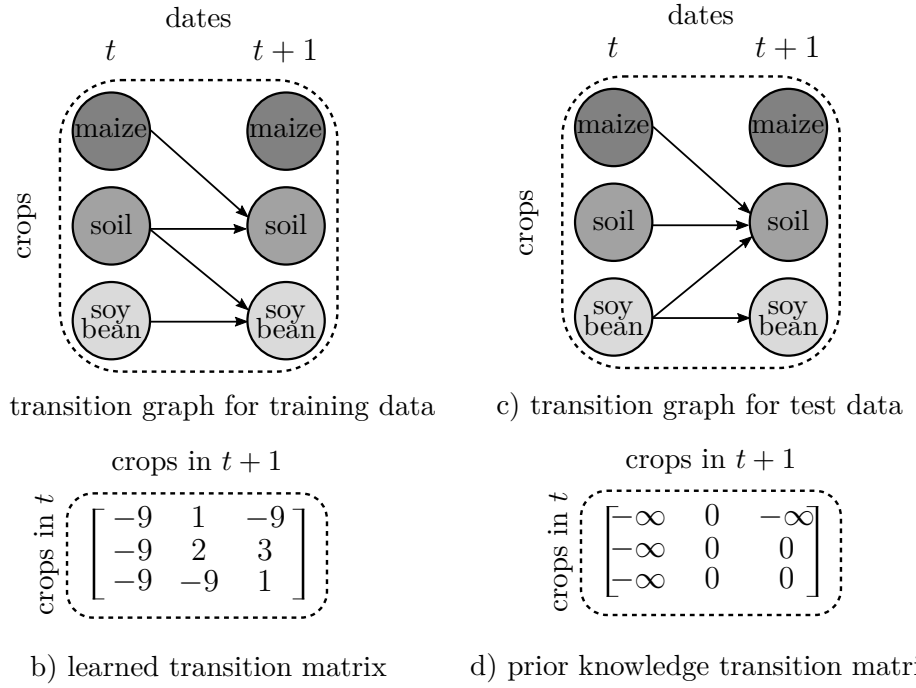

Figure 4.2: Example of learned transitions and prior knowledge models for two adjacent acquisition dates. Transitions graph for training and test data ((a) and (c)); (b) learned transition matrix; (d) transition matrix based on prior knowledge.

Considering this, we propose a second schema to exploit prior knowledge about crops' transitions. In this sense, a human expert on local crop dynamics informs the crops' transitions that might or never occur for each pair of consecutive epochs. In this approach, the model admits class transitions that may occur even if not represented in the training set. The transition matrix based on prior knowledge for the hypothetical target site is presented in Figure

4.2d.

4.2.2.1 Fixing transitions

For the first training variant of the second schema, we use a constant transition matrix that prevents those transitions that may not occur between any pair of adjacent epochs. This constraint is represented by a *constant* transition matrix **Tr** whose elements take value 0 for those transitions that may occur and value $-\infty$ for those transitions that may not happen. This way, the model will admit solutions containing any possible class transitions at test time, even if not represented in the training data, while preventing solutions containing transitions that knowingly never occur.

See, for instance, how the prior knowledge can be considered by defining the transition matrix in the hypothetical example depicted in Figure 4.2d. Note that this transition matrix is more flexible and respects the restrictions observed in the target region, which permits the transition between *soybean* $\rightarrow$ *soil* presented in the test set (Figure 4.2c). As discussed before, in practice, these transition matrices can be provided by human experts on crop dynamics of the target sites. Beyond eliminating sequences inconsistent with the prior knowledge, these constraints can potentially improve the per epoch classification accuracy as reported in [5, 53].

Since **Tr** is now a fixed matrix, the NLL loss function computed gradients only with respect to the network's parameters. We call this variant as *CNN-CRF*$_F$. Notice that, despite the predefined transition scores, *CNN-CRF*$_F$ still models the global conditional distribution for the sequence. At inference time, we again use the CNN prediction and the predefined transition matrix as inputs to the Viterbi decoding stage.

4.2.2.2 Penalizing only the impossible transitions

In the *CNN-CRF*$_F$ model, however, we are not exploiting the true power of a CRF model. In its formulation, the possible transitions all receive the same score, 0.0, regardless of the crop type, limiting learning relevant crop dependencies that may appear in the region. For this reason, we test an approach that only *penalizes* the transitions that may not occur and let the model learn the crop dependencies for those transitions that are possible. We call this variant as *CNN-CRF*$_P$. For this experiment, we start from a predefined transition matrix that adds constraints to those transitions that may not happen between any pair of adjacent epochs. Hence, we propose a training

strategy that initializes the transition matrix with a known **Tr** matrix and lets the network learn the possible transitions. We notice that if the impossible transitions are low enough, i.e., $-\infty$, the gradients for these transitions will be zero and, therefore, will remain unchanged while the possible transitions will be updated. We confirmed experimentally that such a simple strategy allows the model to learn crops' dependencies and maintain a value close to $-\infty$ for the transitions that may not occur. At inference time, we employ the network prediction and the learned transition scores as inputs to the Viterbi algorithm.

4.2.3 Multi-loss learning

Datasets available to train sequence models are often small. This can be even more challenging for agricultural applications since collecting samples for crop mapping is particularly costly, time-consuming, and requires specific domain expertise. However, there is a way to mitigate this problem by jointly training the model to solve multiple tasks. In this sense, multi-loss learning has been widely used for improving model generalization, including sequence tagging problems [56]. When a CNN is trained to perform multiple tasks, the complementary task introduces an inductive bias into the learning process [113], which leads the model to learn features capable of explaining both tasks and therefore generalize better.

Most of the time, in a multi-loss learning scheme, we are interested in one of the tasks. In our case, we are training our model to maximize the sequence labeling task; however, our main objective is to improve the per-date accuracy. Therefore, training our approach to maximize only the sequence could lead to low performance when analyzing the prediction for each independent date since the crop types that maximize the sequence labeling do not necessarily maximize the per-date classification.

For our multi-loss learning scheme, the loss function is defined as the linear combination of two task-specific losses, one associated with the sequence labeling (i.e., the CRF-based loss) and the other considering the per epoch classification disregarding any dependency among the labels. As the second loss function, we employ the per-date categorical cross-entropy. Then, the following join function is applied as the objective function for our multi-loss sequence model:

$$\begin{aligned}\mathcal{L}_{tot} &= -\lambda\mathcal{L}_{crf} - (1-\lambda)\mathcal{L}_{cross} \\ &= -\sum_{i=1}^{A}\sum_{j\in\Omega_i}\left(\lambda \log p(\mathbf{y}^j|\mathbf{x}^j) + (1-\lambda)\frac{1}{T}\sum_{t=1}^{T}\sum_{s=1}^{S}\bar{\mathbf{y}}_{t,s}^{j}\log p(y_{t,s}^{j}|\mathbf{x}^j)\right)\end{aligned} \tag{4-4}$$

where $\mathcal{L}_{crf}$ is the crf loss function (Equation 4-3), $\bar{\mathbf{y}}_t^j$ is the one-hot vector of the true labels for date t, and λ is a weight parameter that controls the contribution of each loss function.

For the multi-loss setting we train the three above-mentioned variants (*CNN-CRF*$_L$,*CNN-CRF*$_F$ and *CNN-CRF*$_P$) using $\mathcal{L}_{tot}$. We call the multi-loss models as *MCNN-CRF*$_{(.)}$, where (.) refers to L, F, or P.

Semantic segmentation with non-dense annotation For training, an FCN usually requires a large number of densely annotated ground-truth samples, which is costly and time-consuming, especially for RS applications. To work around this problem, we propose a modified loss function that enables training with scarce data labeling. To this aim, we used the definition of partial loss [94, 95]. Given a training sample i with partial annotation, the partial loss function only back-propagates the losses from pixels j that belong to the annotated set Ω_i where $|\Omega_i| \neq WH$.

4.3 Baseline models

As baseline models, following [5, 53], we propose two settings. For the first one, we train a CNN that delivers per epoch class scores; we called this model as *CNN* hereafter. For the second setting, we apply the Viterbi algorithm as a post-processing step over the predictions of the *CNN* model using the predefined **Tr** matrix, denoted henceforth as *CNN-Vit*. Different from the *CNN-CRF* variants, this CNN is trained using the categorical cross-entropy loss function. This loss function first applies a softmax function over the network's output that produces a class probability distribution for each time step and then computes the negative log-likelihood over this distribution. Note that in such a way, the classification decision at each time step is conditionally independent of its neighbors. The method can be derived from the general framework in Figure 4.1 by dropping the CRF module. At inference time, we use the class scores predicted by the network and the fixed transition matrices as input to the Viterbi decoding module. This matrix is defined using prior knowledge as in the *CNN-CRF*$_F$ model.

4.4 Accuracy Assessment

The performance of the evaluated methods was expressed in terms of Overall Accuracy (OA) and F1 score (F1). Below is a brief description of these metrics (more details can be found in [114]).

The Confusion matrix records correctly and incorrectly recognized examples for each class. Table 4.2 presents the matrix in mathematical terms. The true classes are noted C_i $(1 \leq i \leq h)$, whereas the estimated classes defined by the classifier, are noted $\hat{C}_j$ $(1 \leq j \leq h)$.

Table 4.2: Mathematical example of confusion matrix.

	C_1	C_2	...	C_h
$\hat{C}_1$	cm_{11}	cm_{12}	...	cm_{1h}
$\hat{C}_2$	cm_{21}	cm_{22}	...	cm_{2h}
...	...	...	...	...
$\hat{C}_h$	cm_{h1}	cm_{h2}	...	cm_{hh}

The terms cm_{ij} $(1 \leq i, j \leq h)$ denote the number of samples recognized as class i in the classification map, when they actually belong to class j in the reference data. Consequently, diagonal terms $(i = j)$ correspond to correctly classified samples and the off-diagonal $(i \neq j)$ terms represent incorrectly classified ones. The sums of the confusion matrix elements over row i and column j are noted cm_{i+} and cm_{+j}, respectively.

The Overall Accuracy (OA) represents the proportion of correctly classified samples with respect to reference data. Thus, OA is a global measure of accuracy, so it depends on larger classes. This measure ranges from 0 (perfect misclassification) to 1 (perfect classification) and can be stated as the trace of the confusion matrix divided by the total number cm of classified instances:

$$OA = \frac{\sum_{i=1}^{h} cm_{ii}}{cm} \tag{4-5}$$

where cm is the total number of elements. The producer's accuracy (PA) value represents the probability that a certain class on the reference is correctly classified. The PA for the class C_j can be computed by:

$$PA_{C_j} = \frac{cm_{jj}}{cm_{+}j}. \tag{4-6}$$

where $cm_{+}j$ is the the summation over all rows i for column j. The user's accuracy (UA) represents the probability that a pixel classified into a given class actually represents that class on the reference. The UA for the class C_i can be computed by:

$$UA_{C_i} = \frac{cm_{ii}}{cm_{i}+}. \tag{4-7}$$

where $cm_{i}+$ is the the summation over all columns j for row i. Finally, the F1 score (F1) is the harmonic mean of UA and PA. F1 is usually more useful than accuracy, especially if the class distribution is uneven. The F1 measure for the class C_i can be computed by:

$$F1_{C_i} = 2 \times \frac{PA_{C_i} \times UA_{C_i}}{PA_{C_i} + UA_{C_i}}. \quad (4\text{-}8)$$

5
Experimental analysis

In this chapter, we describe the experiments to evaluate the multitemporal crop classification frameworks introduced in Chapter 4. Section 5.1 presents the datasets used in the experiments and the training-testing sample selection strategy. Section 5.2 describes the implementation details. Finally, Section 5.3 discusses the results.

5.1
Datasets and study sites

The experiments relied upon two datasets characterized by complex crop dynamics typical of a sub-tropical environment.

5.1.1
Campo Verde

A municipality from Brazil, Campo Verde in Mato Grosso state, was selected to evaluate the proposed methods in tropical regions. Campo Verde dataset is a public dataset available in IEEE DataPort at `https://ieee-dataport.org/documents/campo-verde-database`.

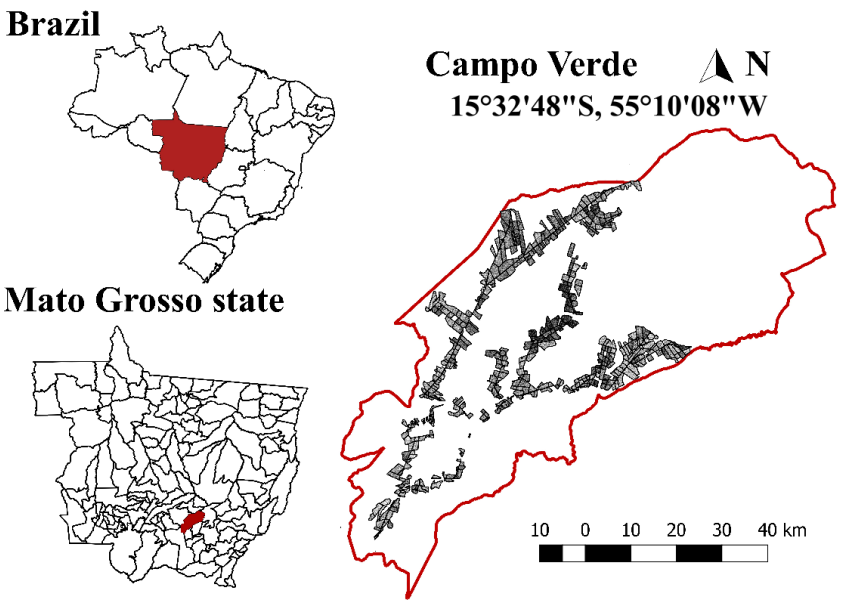

Figure 5.1: Campo Verde, Brazil ([4]).

The study area is situated in the central west region of Brazil (15°32′48”S, 55°10′08”W) (see Figure 5.1). The average annual precipitation is 1726 mm and the average annual temperature is 22.3°C. The major crops found in the area are *soybean*, *maize* and *cotton*. Some minor crops also found the region are *beans* and *sorghum*. In addition, there are *non-commercial crops* (NCC) that includes three classes *millet*, *brachiaria* and *crotalaria*. Other non-crops classes are also present in the dataset, such as *pasture*, *eucalyptus*, uncultivated *soil* (e.g., bare soil, soil with weeds, soil with crop residues), *turfgrass* and *cerrado* (Brazilian savanna). The site covers an extension of 4,800 km^2 approximately with an altitude of 736 m [4]. The dataset comprises a set of of 14 pre-processed SAR Sentinel-1A images, and the reference data (ground truth) for a total of 513 fields (∼6 million pixels). The SAR images are dual polarized (VV & VH), and were captured from October 2015 to July 2016 (see Table 5.1). To add more temporal information at the beginning of the sequences we include the images from October 4^{th} that was downloaded and preprocessed using the same pipeline employed for the other images in the dataset.

As described in [4], the images were acquired from the Sentinel Scientific Data Hub, in Level-1 ground range detected (GRD), and pre-processed using the Sentinel-1 Toolbox. First, a radiometric correction was performed, using the calibration coefficients provided with the Sentinel Level-1 products. Then, a range Doppler terrain correction was applied using a Shuttle Radar Topography Mission (SRTM) digital elevation model (DEM). Next, the VV and VH bands in linear scale were converted to dB. The bands were stacked to form single images, which were then georeferenced to the UTM projection (Zone 21S) and WGS84 Datum, and resampled to 10 m spatial resolution.

Table 5.1: Sentinel-1 acquisition dates over Campo Verde region.

Year	**Month**	**Date**
2015	October	05, 29
	November	10, 22
	December	04, 16
2016	January	21
	February	14
	March	09, 21
	May	08, 20
	June	13
	July	07, 21

The agricultural practices in the region consist of two seeding periods for the major crops, *soybeans* span from October to February, and *maize* and *cotton* from Mars to July. The phenological cycles of the main crops

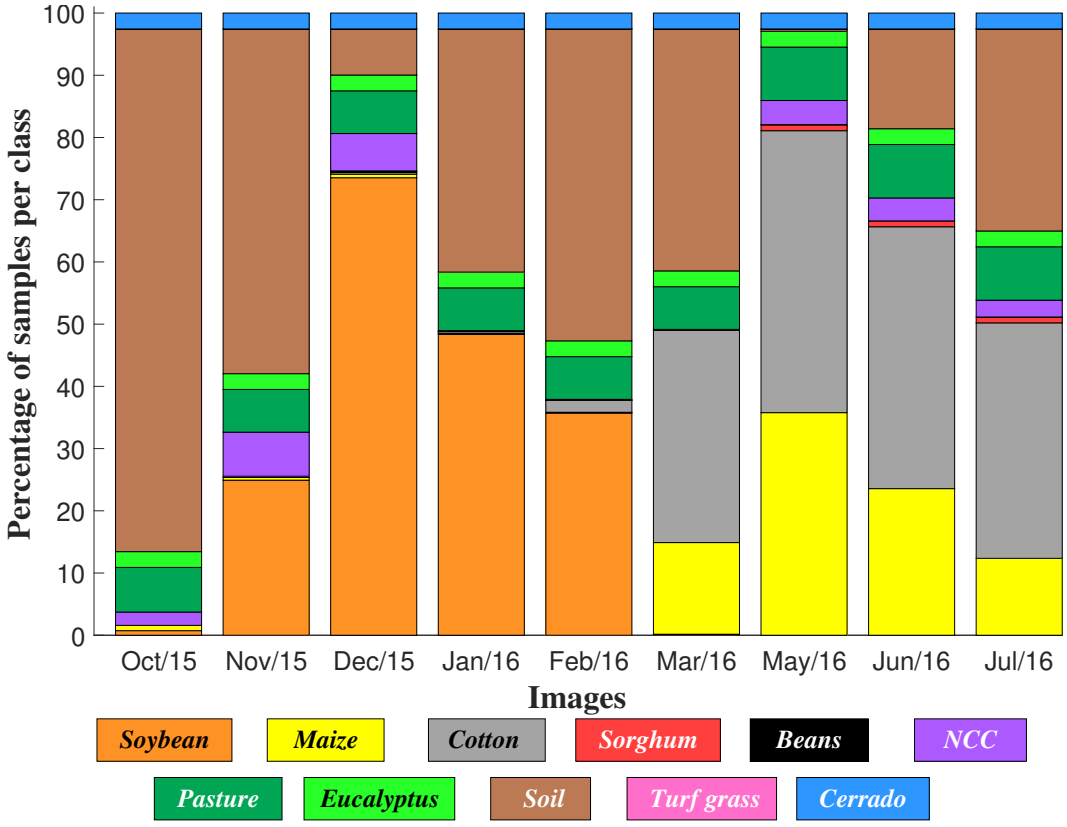

Figure 5.2: Class distribution in Campo Verde dataset ([4]).

can span 3 to 4 months (*soybeans* and *maize*) and 4 to 6 months (*cotton*). Figure 5.3 shows the crop calendar for the major crops and illustrates how complex the crop dynamics is in this region. Some crop rotations present in the dataset are *soybeans-maize*, *soybeans-cotton*, *soybeans-sorghum*, *soybeans-pasture*, *soybeans-beans*, *beans-cotton* and *maize-cotton*. Figure 5.2 shows how the area is distributed among different crops along the months. The graph shows that the cycle of the same crop, e.g., soybean or maize, does not start in the same month in all fields and its duration can also vary from one field to another. As mentioned before, such complex crops' dynamics are characteristic of tropical regions.

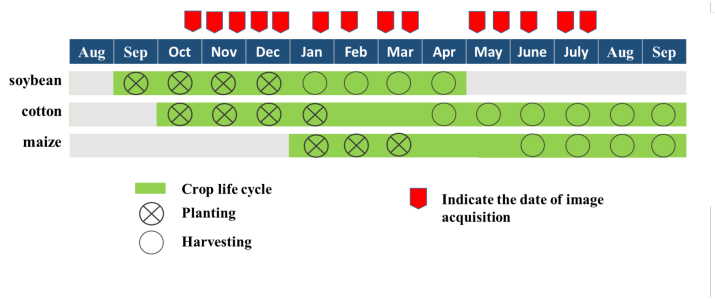

Figure 5.3: Crop calendar for major crops in Campo Verde.

The original reference data consists of 513 crop fields, but to produce

training and validation sets with at least one field for each class, some fields were split up, thus generating a total of 608 fields. To avoid that pixels from the same field could fall in the training and testing sets, the selection was performed at the crop field level. Two disjoint sets of fields were then selected, one for training and the other for testing, using stratified random sampling. Approximately 50% of the polygons of each class were selected for training and 50% for testing.

5.1.2 Luis Eduardo Magalhães

The second test site is in Luis Eduardo Magalhães (LEM) municipality, Bahia state, Brazil, with an area of 3,940 km^2 [115]. It is at a latitude of 12°05'31" south and longitude 45°48'18" west (see Figure 5.4). The average temperature in the region is 24.2 °C and the average annual rainfall is 1511 mm. Similar to Campo Verde dataset, the class distribution in LEM dataset is non uniform along the year, as shown in Figure 5.5. The main crop types are *soybean*, *maize*, *cotton* and *millet*. Some minor crops also found in the region are *beans*, *coffee* and *sorghum*. Other land use classes are: *pasture*, *eucalyptus*, *hay*, *grass*, uncultivated *soil*, and *cerrado* (Brazilian savanna). The LEM database is freely accessible at `http://www.dpi.inpe.br/agricultural-database/lem/`.

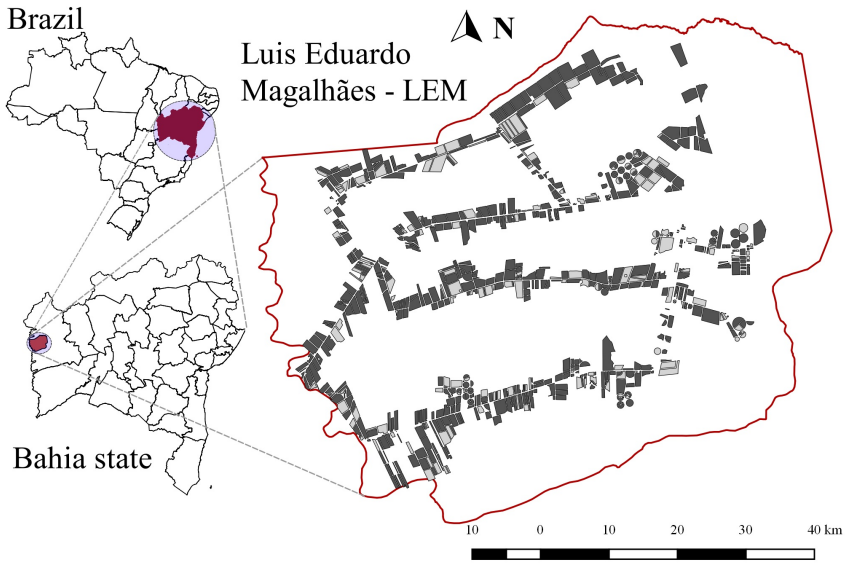

Figure 5.4: Luis Eduardo Magalhães (LEM), Brazil ([5]).

As described in [115], the Sentinel-1A images with VV and VH polarizations were acquired from the Sentinel Scientific Data Hub in Interferometric Wide Swath (IWS) mode (Ground Range Detected (GRD) Level 1 product) and were pre-processed using the Sentinel-1 Toolbox 5.0. The pipeline consists

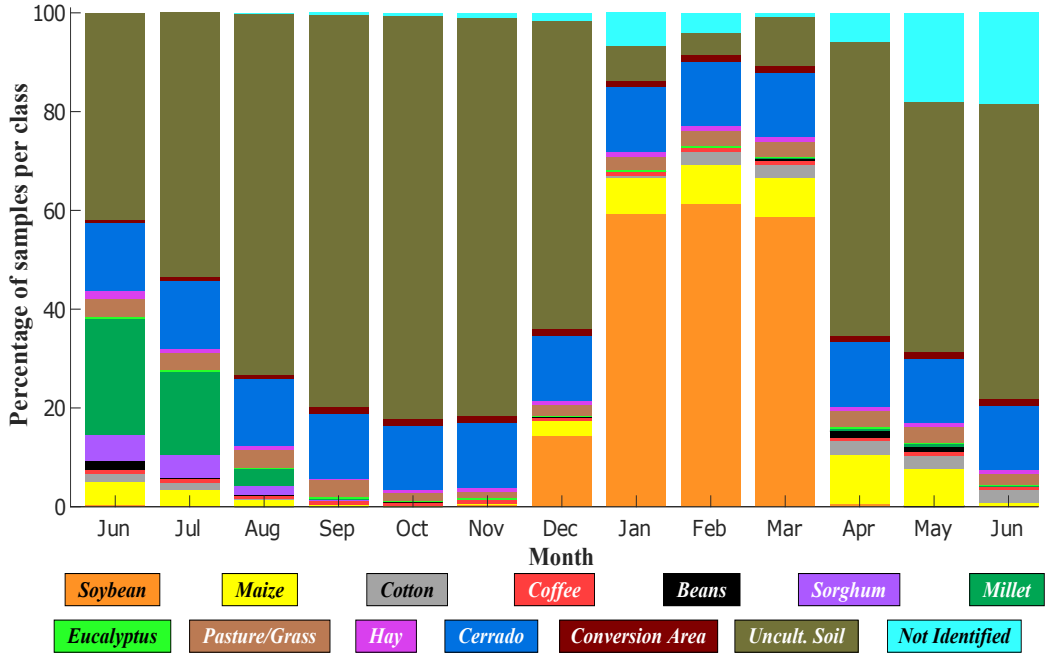

Figure 5.5: Class distribution in LEM dataset ([5]).

of applying orbit file, radiometric calibration, terrain correction, and linear transformation to dB. The SAR images are dual polarized (VV & VH), and were captured from June 20157 to June 2018 (see Table 5.2).

Table 5.2: Sentinel-1 acquisition dates over LEM region.

Year	Month	Date
2017	June	12, 24
	July	06, 30
	August	11, 23
	September	04, 16, 28
	October	10
	November	03, 15, 27
	December	09, 21
2018	January	02, 14, 26
	February	07, 19
	March	03, 15, 27
	April	08, 20
	May	02, 14, 26
	June	07, 19

The original reference data consists on 794 crop fields, obtained in two field campaigns between 26-30th June 2017 and 14-19th March 2018, periods corresponding to the second and first Brazilian crop harvests, respectively. To produce training and testing sets with at least one field for each class, some fields were split up, thus generating a total of 807 fields. To avoid that pixels

from the same field could fall in the training and testing sets, the selection was performed at the crop field level. Using stratified random sampling, two disjoint sets of fields were then selected for training and the other for testing. Approximately 75% of the polygons of each class were selected for training and 25% for testing.

5.2 Experimental Design

5.2.1 Experimental Protocol

All CNN-CRF variants take as input an image patch and compute the class posterior probabilities for all pixels within the patch. Patches from the original images were selected as primary input features. We trained and validated our models using image tiles of size $Nb \times 128 \times 128 \times 2$, where Nb is the sequence of input images, 15 for Campo Verde, and 30 for the LEM dataset. To extract the tiles, we used a random-crop strategy with the following pipeline. Firstly, we used a regular grid to sample the tiles' central coordinates, setting a grid spacing to derive 95% of overlap between neighboring tiles and create sufficient training samples. Secondly, we cropped square tiles of $Nb \times 128 \times 128 \times 2$ from the orthoimage and the digitized polygons. Using this strategy, we randomly cropped $50,000$ image tiles on the fly at each epoch for both datasets, guaranteeing that at least 10% of each tile is annotated with a reference field.

As reported in Figures 5.2 and 5.5, some crop types outnumber others by a large margin in the number of annotated pixels. Class imbalanced datasets significantly reduce deep learning models' accuracy, creating a bias to learn those features that best discriminate among the classes with the higher number of samples. Thus, we proposed a strategy to ensure that all classes have similar probabilities of appearing in a cropped tile by oversampling the under-represented classes. We implemented an image tiles selection process that produces multiple views (total or partial) of the same field, operating a data augmentation process. It is worth pointing out that this strategy does not ensure the same number of labeled pixels per class.

5.2.2 Fully convolutional architecture design

Figure 5.6 and Table 5.3 depict the CNN-CRF architecture design of our approach. The network consists of an encoder that learns general low-

level features and a decoder that recovers the spatial resolution and performs classification. The network is trained end-to-end using the loss functions described in the previous chapter.

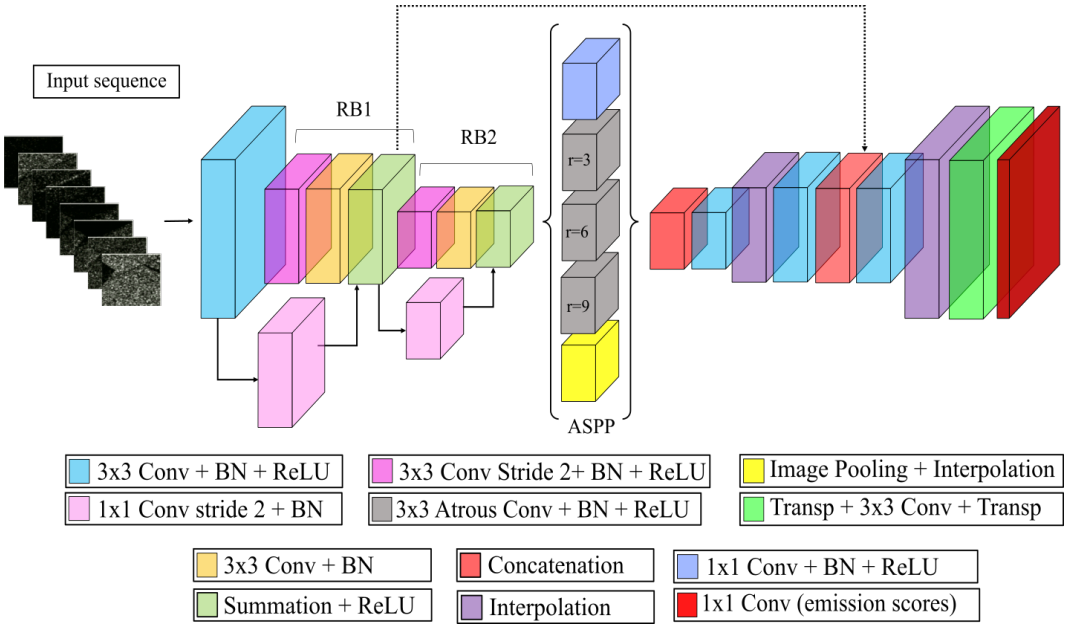

Figure 5.6: 3D Fully convolutional Network architecture. It consists of a ResNet-based encoder combined with a DeepLabv3+-based decoder that delivers the emission scores.

The encoder was designed based on the ResNet architecture [110]. Even though residual blocks enable deeper architectures, we experimentally set up a network called ResNet-7, composed of only seven convolutional operations in 2 residual blocks (RB1 and RB2 in Figure 5.6). In an exploratory test, we found that this shallow configuration provided a better trade-off between computational cost and accuracy than standard versions of ResNet, including ResNet-18 and ResNet-50. Each residual block is composed of 3D convolutional layers (with spatial context of 3×3 or 1×1 according to the operation, see Figure 5.6), 3D Batch Normalization operation, and ReLU activation function. The spatial dimension was reduced by using convolution operations with stride 2. Unlike ResNet architecture, we used a first convolutional block (CB1) that did not reduce the spatial dimension. The number of filters doubles periodically at each residual block, and the spatial resolution of the output feature maps is two times smaller than the input resolution. To recover the spatial dimension, we feed the ResNet output feature map to a decoder based on the DeepLabv3+ architecture [109]. Similar to [109], our ASPP module consists of 5 parallel operations: an image pooling, a 1×1 convolution, and three 3×3 atrous convolution with r equal 3, 6 and 9, respectively. We concatenated the five outputs and used two convolutional blocks (CB2 and CB3) consisting

Table 5.3: Architecture of the *CNN-CRF* network.

Layer	**Processing**
Input	2×Nb×128×128
CB1	62×Nb×128×128
RB1	64×Nb×64×64
RB2	128×Nb×32×32
ASPP (1-5)	64×Nb×32×32
Concat	320×Nb×32×32
CB2	64×Nb×32×32
Interp	64×Nb×64×64
CB3	64×Nb×64×64
Concat	128×Nb×64×64
CB4	64×Nb×64×64
Interp	64×Nb×128×128
SetTemp	64×T×128×128
Emission Conv	S×T×128×128
Tr	(T-1)×$\|S\|$ × $\|S\|$

of a 3D convolution followed by a BN, a ReLU activation function, and a bilinear upsampling to recover the input spatial resolution. We also use skip-connections by concatenating the CB2 output with the corresponding encoder low-level features (depicted in Figure 5.6 with the dotted black lines). All the above-mentioned 3D convolutions consider a temporal context of 5 months, and we employed temporal padding to preserve the temporal dimension. Sometimes, as in our datasets, one can have several images per epoch/month; however, we are interested in a unique prediction for each month. Hence we implemented a convolutional processing layer (penultimate layer in Figure 5.6 and the SetTemp layer in Table 5.3) that maps the temporal dimension to the desired length. Finally, a last convolutional layer with linear activation with $|S|$ kernels of size $1 \times 1 \times 1$ delivers the emission scores for each class for each month of interest.

The second term of the CRF model (second term in Equation 4-1) is implemented by defining the transition matrix of size $(T-1) \times |S| \times |S|$ (**Tr** in Table 5.3), as discussed in Chapter 4. Depending on the variant, this matrix will be either learned or considered a constant. Regarding the temporal dynamic, the transition matrix for the *CNN-CRF*$_F$ and *CNN-Vit* models is derived from the reference data (including both training and test data). We employed the references data considering that it represents the crops' dynamics for the target region. Even though we employed the reference data in our experimental setup, it is worth pointing out that in a real operational case, these matrices can be given by an expert who provides the information about the transition that

may or may not occur between adjacent months during the agricultural year.

To train our models, we set the less probable transitions to -5. In preliminary experiments, we varied the transitions between -10 and -1, and -5 delivered the best results. In addition, for *CNN-CRF$_F$* and *CNN-Vit* models, we assumed that all transitions at the beginning and end of the sequence had the same score. For the others CNN-CRF variants, the start and end transitions were learned from the training data.

At inference time, we applied the trained network and the Viterbi algorithm to overlapping image tiles using a sliding window strategy with 30% of overlap and keeping the patch's central region, minimizing border effects. Finally, we concatenated the outputs to obtain an outcome with the input image dimensions.

All models were trained using stochastic gradient descent (SGD), with a weight decay of 1e−6 and an initial learning rate of 0.1. We warmed up the learning rate during one epoch and then used the cosine learning rate decay [116] with a final value of 0.0001. We trained for 50 epochs with 16 image patches per batch, monitored the average F1 score, and applied early stop when no improvements higher than 0.9*e*e−4 happened in a sequence of 10 epochs. We ran each experiment five times, using five different seeds. All experiments were carried out on an Intel Core i7-4790, 32Gb RAM, and a GPU NVidia GeForce Titan GTX 1080 (11Gb RAM).

5.3 Results and Discussion

In this section, we report the accuracy assessment for all CNN-CRF variants considering the test set in both Campo Verde and LEM datasets. First, we present and discuss the results for the *MCNN-CRF$_P$* variant, which reported the best performance on both datasets. Then, we analyze how each proposed variation of the general CNN-CRF model contributes to the final classification performance. Finally, we discuss the limitations of the methodology.

5.3.1 Results for Campo Verde dataset

5.3.1.1 Quantitative results

Best CNN-CRF variant compared with the baseline approach: As expected, the combination of multi-loss learning - impossible transition penalization - possible transition learning, i. e. *MCNN-CRF$_P$* variant delivered the

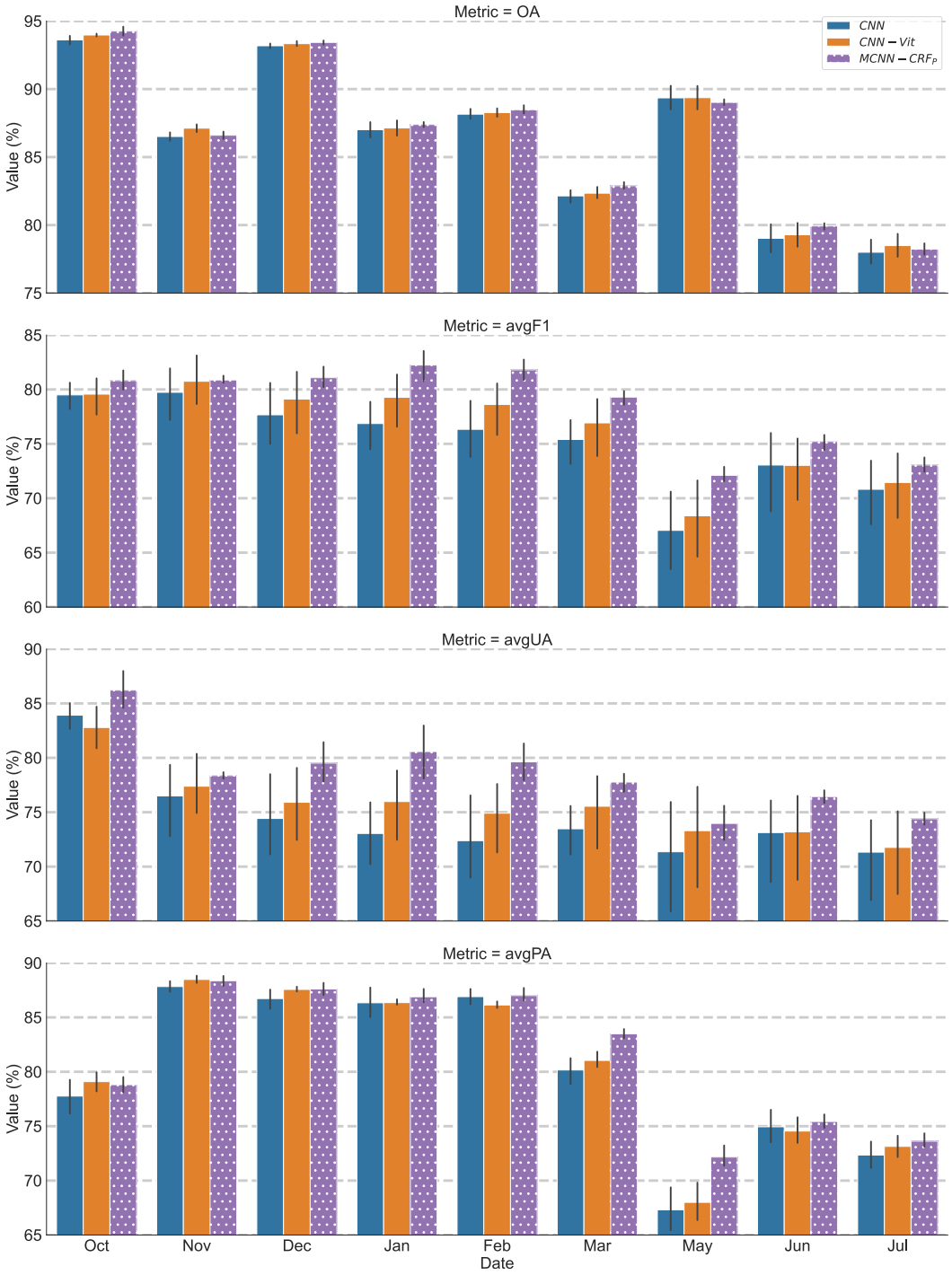

Figure 5.7: Overall Accuracy (OA), average F1 score (avgF1), average producer's accuracy (avgPA), and average user's accuracy (avgUA), computed each month for Campo Verde dataset for *MCNN-CRF$_P$* model and the baseline models. Values are the average over five runs, with the black line indicating each model's minimum and maximum value.

best results compared to the baseline approach. Figure 5.7 summarizes the per-month results obtained for *MCNN-CRF$_P$* in terms of OA, average F1 score (avgF1), average producer's accuracy (avgPA) and average user's accuracy (avgUA). The horizontal axis contains the month being classified. In

this figure, we report just one result per month, considering the nine annotated months for the Campo Verde dataset. In addition, the figure presents the results for the baseline model where we report the performance for the CNN output trained solely with a per-month categorical cross-entropy (first bar in the figure) and the performance after applying the Viterbi algorithm to the CNN output (*CNN-Vit* in the figure).

Figure 5.7 provides a clearer view of the relative performance of the tested methods. As expected, applying the Viterbi decoding (*CNN-Vit*) improved the CNN output according to all metrics in almost all months. Furthermore, the results revealed that *MCNN-CRF$_P$* delivered definite improvements compared to the baseline method *CNN-Vit* in terms of avgF1, avgUA, and avgPA. In particular, our approach is more robust to the network initialization, reporting slight performance variations among the five runs.

In terms of OA, no relevant differences were observed among both methods; nonetheless, *MCNN-CRF$_P$* was superior to *CNN-Vit* in six out of the nine months by a low margin. In contrast, it reported a drop in performance with a shallow margin for the other three months. These results indicate that our proposal improved the classification principally for minority classes.

Scrutinizing the F1 score for each crop type individually, we observed the most significant gains in performance for classes *sorghum*, *turfgrass* and *beans*. Figure 5.8 reports the improvements/drops in F1 per class brought by both models for months from December to May, for which the higher differences were observed between the baseline and the proposed model. In the figure, the horizontal axis contains the crop being classified, and the vertical axis contains the F1 score improvements/drops in percent for *CNN-Vit* and *MCNN-CRF$_P$* with respect to the *CNN* model. As observed, *MCNN-CRF$_P$* delivered significant improvement for the above-mentioned crop types, reaching up to 30% for class *sorghum*, 23% percent for class *turfrgass*, and 28% for class *beans*. However, the opposite occurred in December for class *beans*, where *MCNN-CRF$_P$* reported a drop of 6.49% in F1 score due principally to a low value in the user's accuracy. Nonetheless, the *MCNN-CRF$_P$* model improved the F1 score for almost all classes across the whole sequence.

For a better understanding of these variations on the F1 score, Table 5.4 presents some examples of training sequences presented in the dataset. In the table we can observe that crop rotations *soybean-cotton* and *beans-cotton* (first and second row in the table) both have the same temporal sequence starting from January. In addition, the temporal sequence for rotation *beans-cotton* is unique in the training set, whereas, due to displacement in time, there are others possible sequences with rotation *soybean-cotton* (see Table 5.4

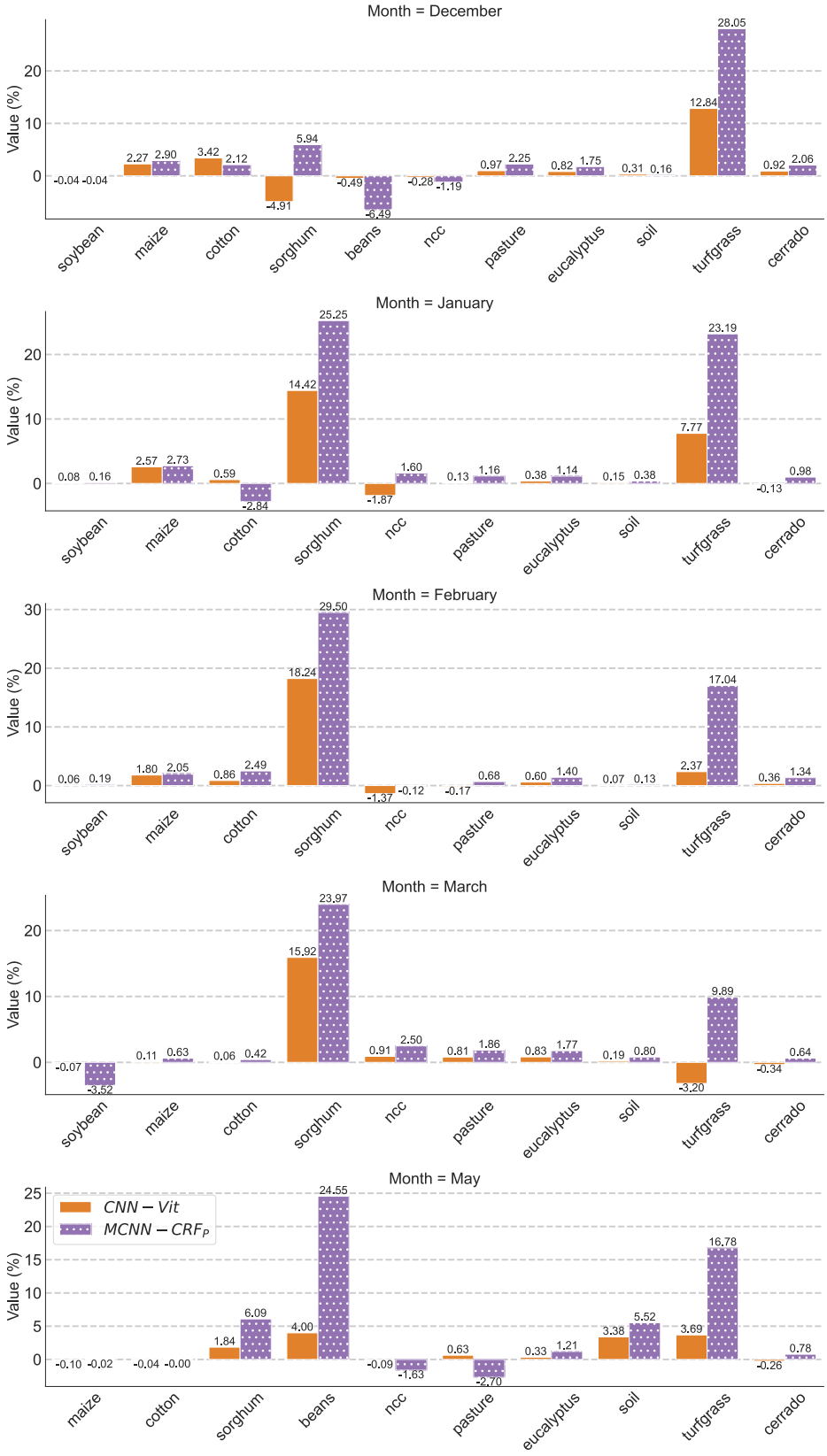

Figure 5.8: F1 score improvements/drops for *CNN-Vit* and *MCNN-CRF$_P$* with respect to *CNN*, from January to May (from top to bottom). Campo Verde dataset.

Table 5.4: Examples of training sequences for Campo Verde. Pixl. stands for the number of training pixels for each sequence.

Pixl.	Oct	Nov	Dec	Jan	Feb	Mar	May	Jun	Jul
8K	*soil*	*beans*	*beans*	*soil*	*soil*	*cotton*	*cotton*	*cotton*	*soil*
63K	*soil*	*soybean*	*soybean*	*soil*	*soil*	*cotton*	*cotton*	*cotton*	*soil*
109K	*soil*	*soil*	*soybean*	*soybean*	*soybean*	*soil*	*cotton*	*cotton*	*cotton*
257K	*soil*	*soil*	*soybean*	*soybean*	*soybean*	*soil*	*maize*	*maize*	*soil*
224K	*soil*	*soil*	*soybean*	*soybean*	*soybean*	*soil*	*maize*	*maize*	*maize*
1K	*soil*	*soil*	*soybean*	*soybean*	*soybean*	*soil*	*beans*	*soil*	*soil*
1.4K	*soil*	*soil*	*sorghum*	*sorghum*	*sorghum*	*sorghum*	*soil*	*soil*	*soil*

3rd row). Furthermore, transition between *soybean-soybean* from December to January are more frequent in the training data (rows 2 to 5 in Table 5.4) that transition *soybean-soil*. Considering this, the possible transitions learned by *MCNN-CRF$_P$* will assign higher score to the transition *beans-soil* between December and January, than the score assigned for transition *soybean-soil* for the same date. Now let's assume that the CNN model has low confidence in discriminating between *soybean* and *beans* and delivers similar emission scores for both crop types in December. Under this condition, the Viterbi algorithm could select the path that contains the higher transition scores from December to January.

To confirm this expectation, we also reported the *MCNN-CRF$_P$* confusion matrix for December (see Figure 5.10) and the transition matrix learned for adjacent months December-January with the less probable transitions set to -5 (see Figure 5.9). As observed, the model assigned a score of 0.00 for transition *beans-soil* and -0.03 for transition *soybean-soil*. Notice that, as we discussed, given the class *soybean* in January, the higher score will be obtained by transitioning to also class *soybean*, which matches the expected transition (see Table 5.4). Considering these transition scores, if the emission scores for both classes is similar, the Viterbi algorithm will deliver the sequence that includes the transition *beans-soil* between December and January, therefore miss-classifying *soybean* as *beans* for December, as reported in Figure 5.10.

In contrast with the drop in performance for crop type *beans* for the first growing season (October to February), we observed a significant gain in terms of F1 score for the second growing season (March to July) (see Figure 5.8). Class *beans* on the second half of the sequence presents a very distinctive temporal constraint with only one month containing this crop type (see Table 5.4 row number 4). The same applies for class *sorghum*, which present a completely different temporal dynamic when compared with the rest of the sequences (see Table 5.4 row number 5); and classes *turgrass* and *cerrado* which are non-crop classes that do not change in time. This could potentially explain the better performance presented by *MCNN-CRF$_P$* compared to *CNN-Vit*.

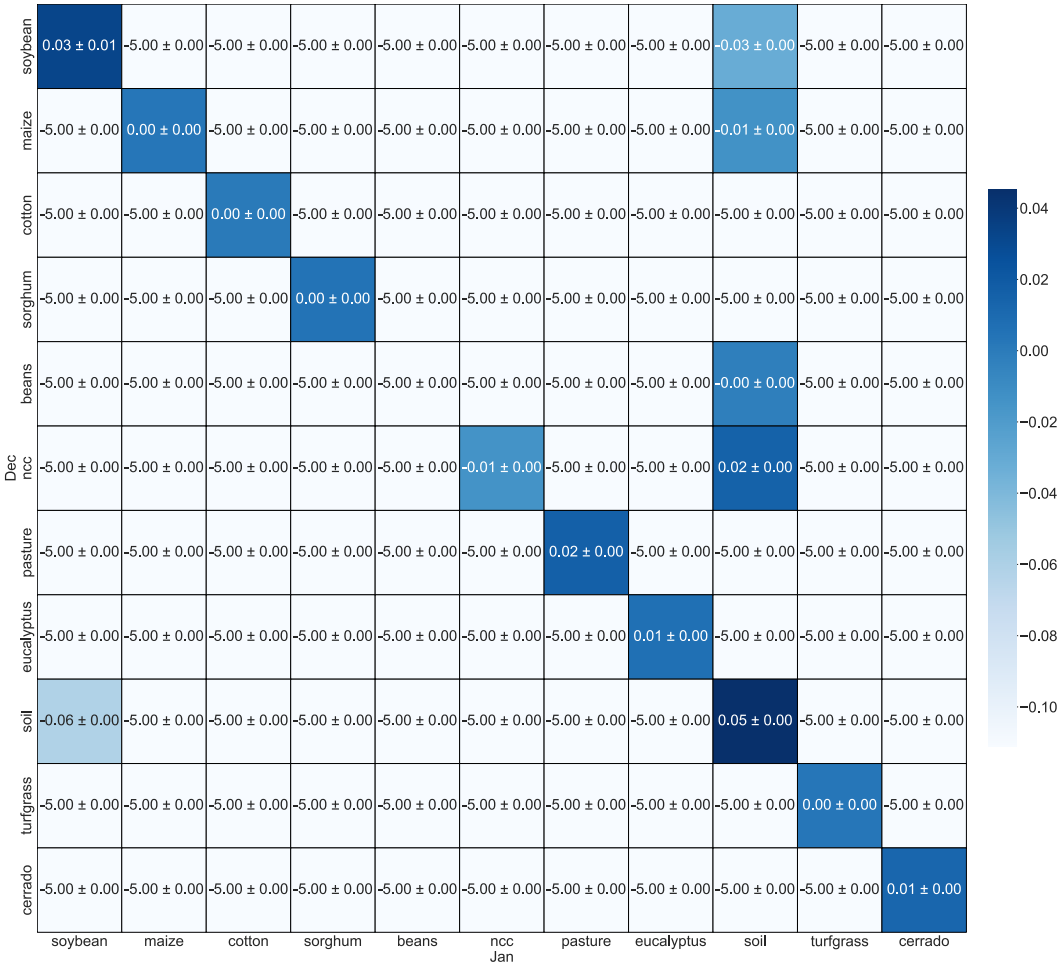

Figure 5.9: Transition matrix for adjacent months December-January learned by *MCNN-CRF$_P$* model on Campo Verde dataset.

Comparison between data-driven and prior-knowledge: Figure 5.11 summarize the per-month results obtained for the proposed variants for schema 1 and schema 2 (see Table 4.1) experiments: fixed transitions based on prior knowledge *CNN-CRF$_F$*; penalizing only the less probable transitions *CNN-CRF$_P$*; learned transitions *CNN-CRF$_L$*. We also report the results for the baseline model.

Considering the variants that employ prior knowledge, we observed that *CNN-CRF$_F$* gains to *CNN-Vit* in five out of the nine months in terms of avgF1, with a high margin in four of them, whereas for the other four months reports a moderate drop in performance. Specifically, *CNN-CRF$_F$* reported better producer's accuracy (avgPA in Figure 5.11) than the baseline approach, and worst results in term of user's accuracy (avgUA in Figure 5.11). Analyzing *CNN-CRF$_P$* variant, we observed that the model delivered lower results compared to *CNN-CRF$_F$*. However, the error bars in the figure also indicate higher robustness for *CNN-CRF$_P$* model, presenting low standard deviation than *CNN-CRF$_F$* for almost all months.

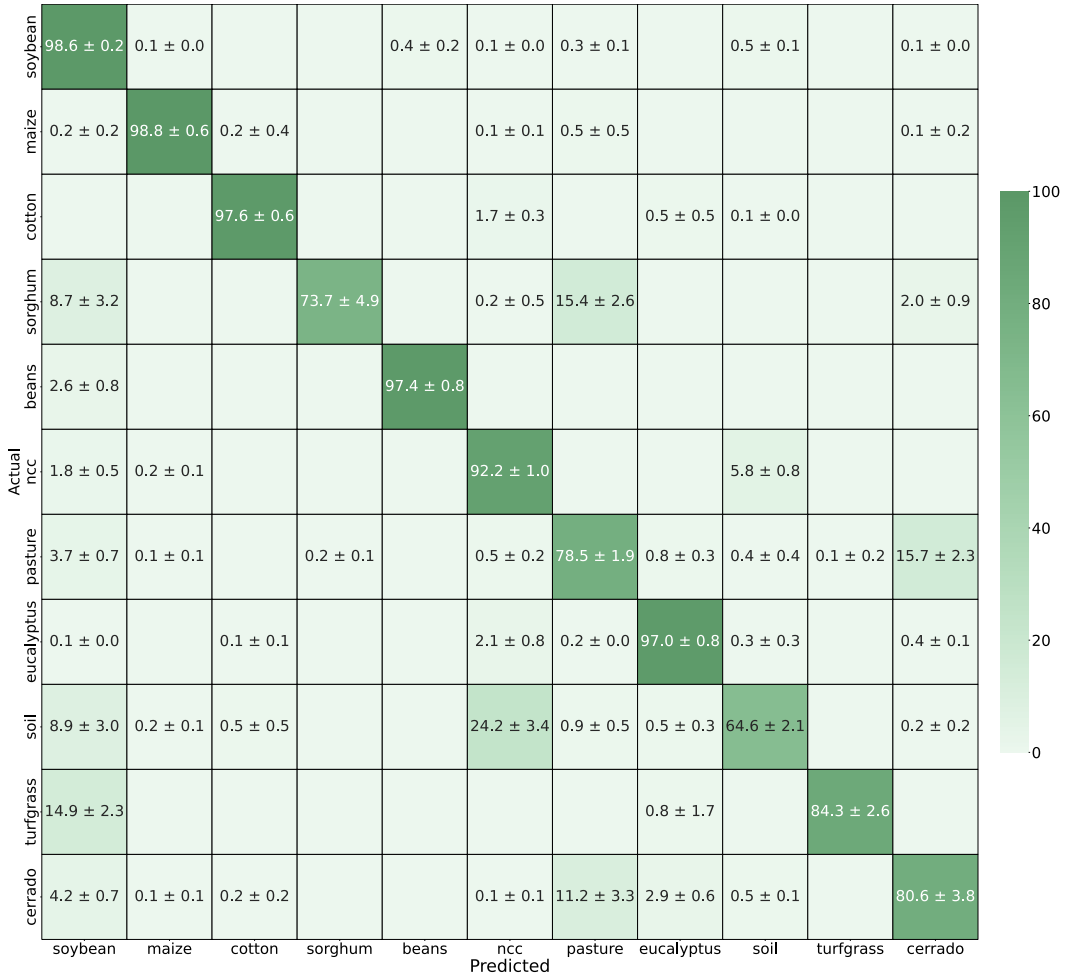

Figure 5.10: Confusion Matrix for month December for *MCNN-CRF$_P$* model on Campo Verde dataset.

Considering the variant that learns the transitions scores, the results revealed that *CNN-CRF$_L$* outperformed all methods in terms of avgF1, avgPA, and avgUA, for nearly all tested months. These results indicate that learning the transition matrix brought major flexibility to the model. We hypothesize that using fixed values for the less frequent transitions in *CNN-CRF$_P$* can be too restrictive when training only with the CRF-based loss. For that reason, we propose the multi-loss schema, which results are discussed in the following section.

Multi-loss learning: Figure 5.12 summarizes the per-month results obtained for the proposed CNN-CRF variants now trained using a multi-loss schema. Comparing the single-loss results for the variants that use prior knowledge (*CNN-CRF$_F$* and *CNN-CRF$_P$*), the corresponding multi-loss results (*MCNN-CRF$_F$* and *MCNN-CRF$_P$*), we found that the inclusion of the per-month cross-entropy loss in our CNN-CRF models brought significant ac-

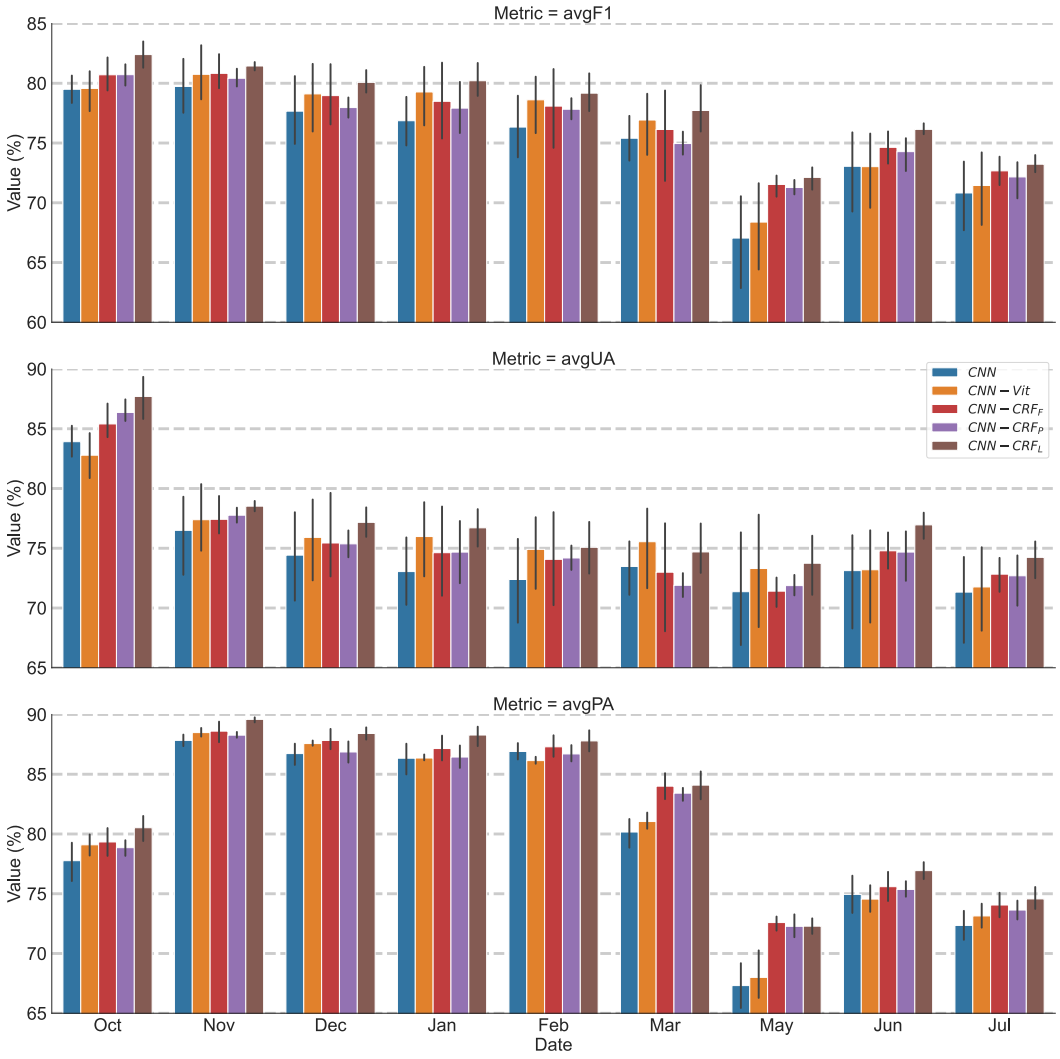

Figure 5.11: Average F1 score (avgF1), average producer's accuracy (avgPA), and average user's accuracy (avgUA), computed each month for Campo Verde dataset for single-loss data-driven and prior-knowledge variants. Values are the average over five runs, with the black line indicating the minimum and maximum value for each model.

curacy gains from December to January. Such improvements were more significant for the *MCNN-CRF$_P$* that consistently outperformed *CNN-CRF$_P$*.

It is worth noticing that *MCNN-CRF$_F$* reported a moderate loss in performance for October, June, and July, due to classes *turfgrass* and *cerrado* which are two of the classes that benefit the most from learning the possible transitions in the *MCNN-CRF$_P$* variant. For the *CNN-CRF$_L$* variant, notwithstanding, the inclusion of the cross-entropy loss brought detrimental results for some dates, reporting moderate increases in the producer's accuracy (PA) and losses in the user's accuracy (UA).

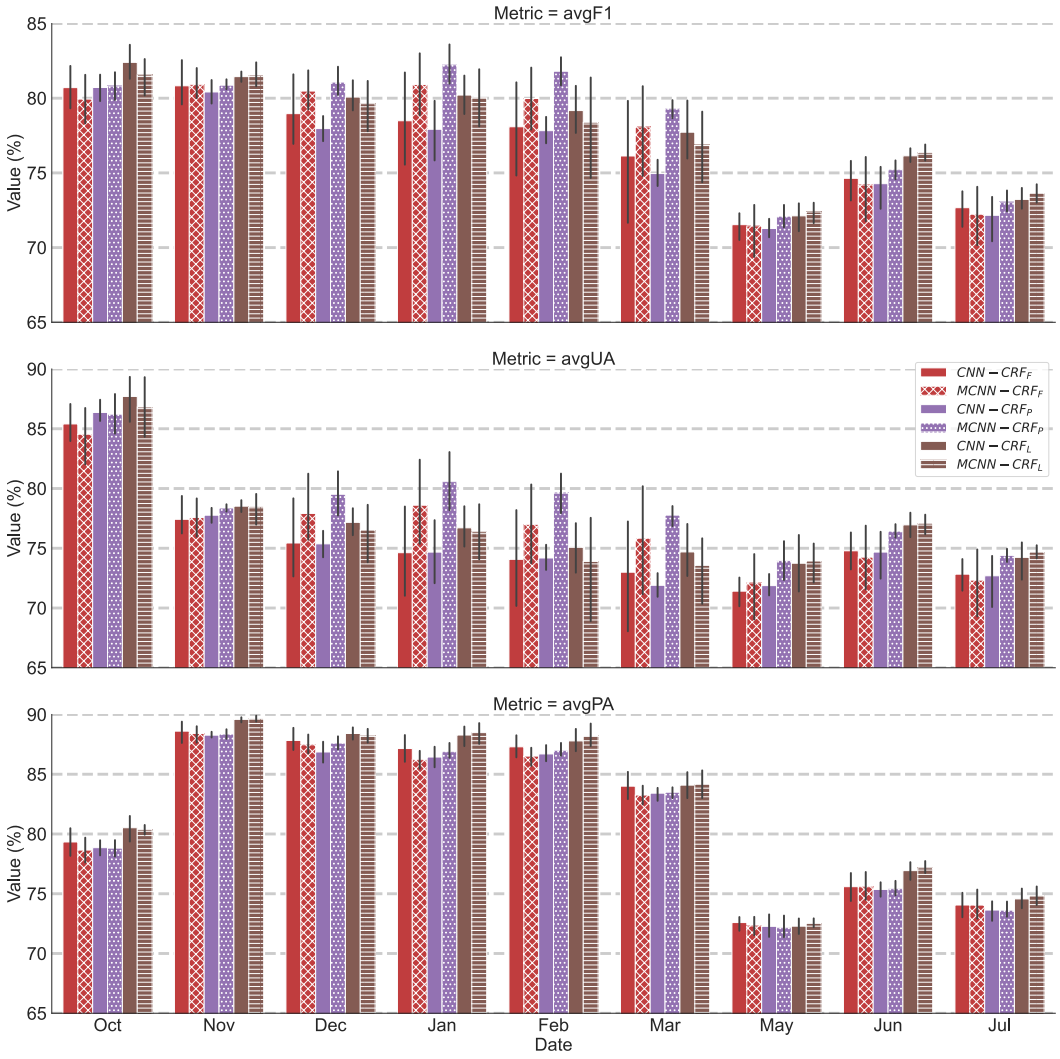

Figure 5.12: Average F1 score (avgF1), average producer's accuracy (avgPA), and average user's accuracy (avgUA), computed each month for Campo Verde dataset for the single-loss and multi-loss hybrid models. Values are the average over five runs with the black line indicating the minimum and maximum value for each model.

5.3.1.2 Qualitative results

Figure 5.13 presents the classification maps for each method in a selected area. For conciseness, we showed the results from January to May. In addition, Figure 5.13 shows the error maps, where the dark orange color represents the misclassified regions. The prediction maps presented good accuracy for most of the classes for all methods. Classification errors were observed mostly in the misclassification of one of the main crop with the class *soil*, for example *soybean* in January and February, and *maize* and *cotton* in March and May. Figure 12 presents examples of the results predicted by the four approaches. These errors occurred quite often in our experiments for seeding and harvest time. Between

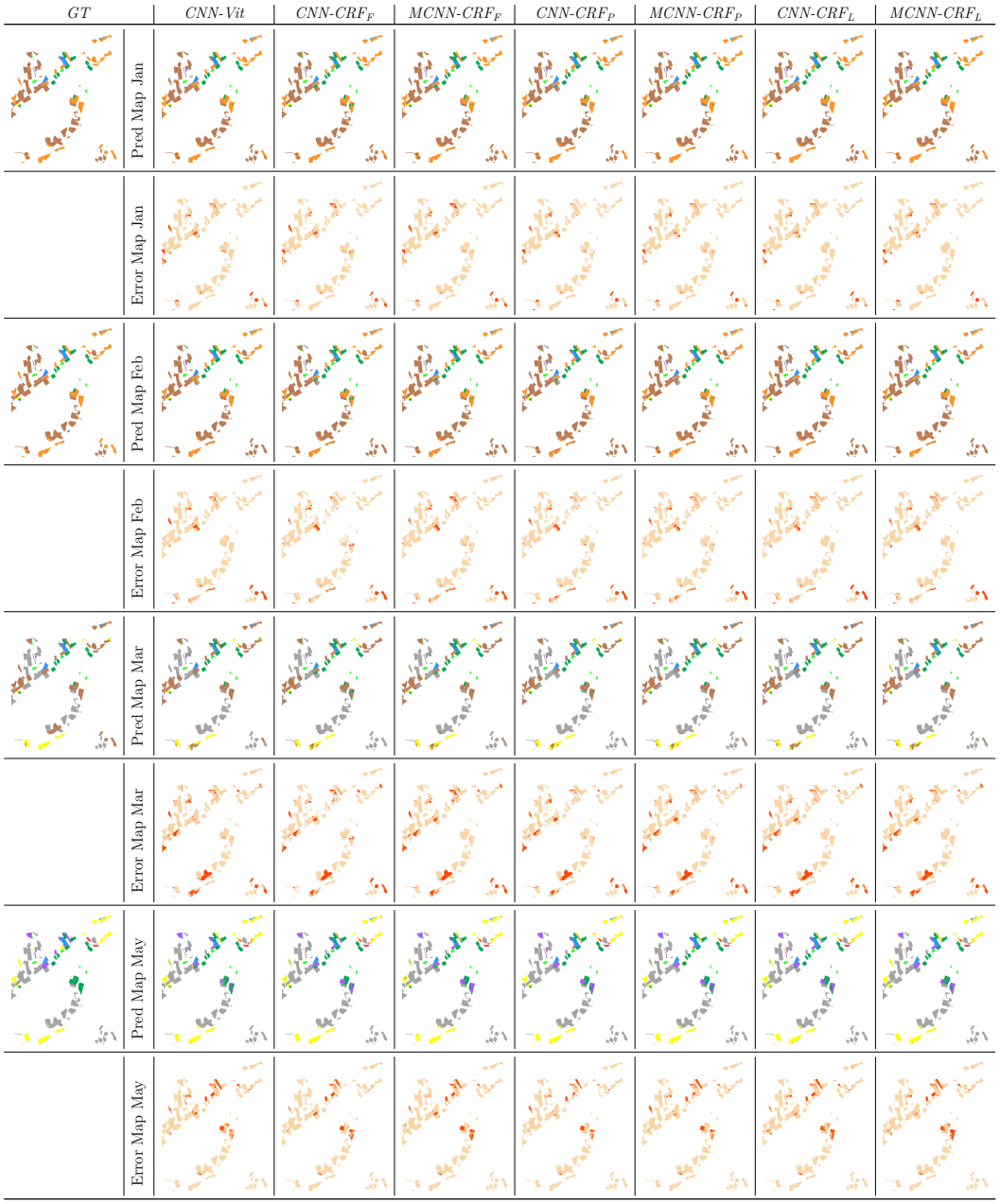

Figure 5.13: Prediction and error maps for each method for selected months for Campo Verde dataset. GT stands for ground truth. The prediction maps use the same color legend as in Figure 5.2. For the error maps, dark orange is the misclassified area.

February and March, it was the *soybean* harvest time and *maize* and *cotton* seeding time; however, due to the agricultural practices in the region, harvest and seeding did not occur on the same date for all parcels. For example, in this period, *maize* and *cotton* were in some parcels in their early growing stage and could easily be confused with soil. A similar problem came about around harvest time. Other type of confusions were observed between *pasture* and *cerrado* for all months; *pasture* and *NCC* for May; and *maize* and *NCC* for May.

Finally, the entropy for the class emission scores for each CRF-based method is presented in Figure 5.14. These maps give insight into the multi-loss models' success that employed prior information about the transitions. As observed, the entropy decreases for both *CNN-CRF$_F$* and *CNN-CRF$_P$* models when using the multi-loss training scheme, which indicates higher confidence in the predicted class. These results showed that using multi-loss training can potentially reduce the classification uncertainty in those models that employ priors, making the classification results more reliable. We also observed that *CNN-CRF$_L$* and *MCNN-CRF$_L$* both reported high confidence in their predictions.

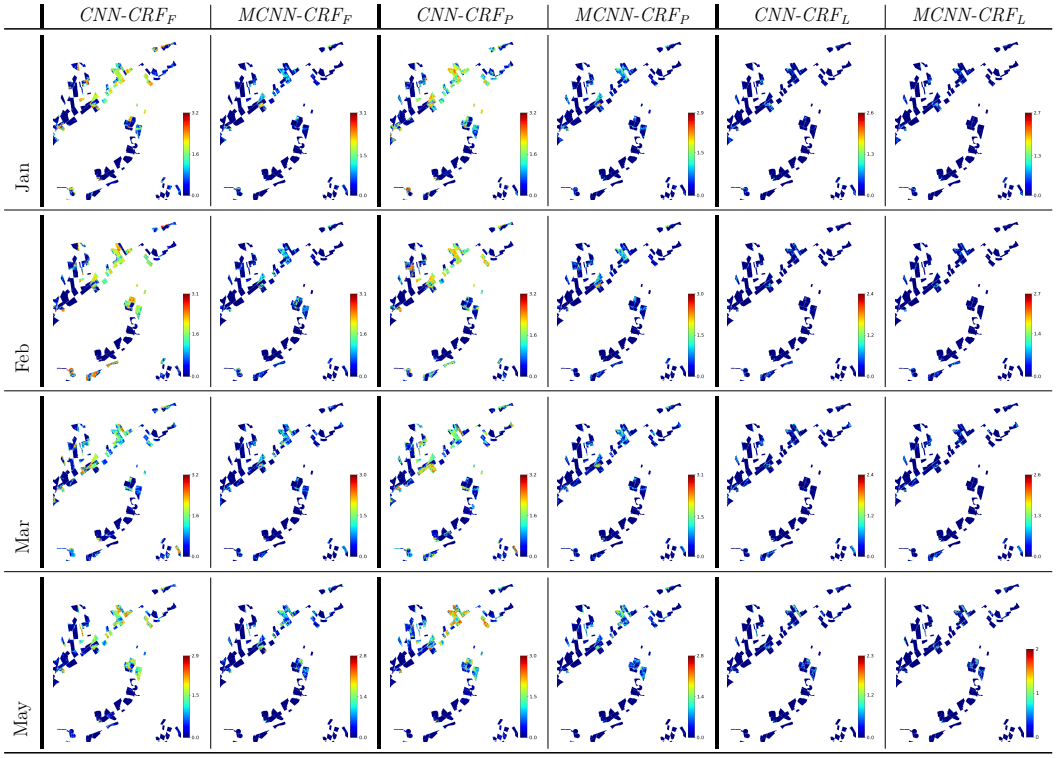

Figure 5.14: Maps of the entropy values for the emission scores for CRF-based models for a selected area for the months January to May for Campo Verde dataset.

5.3.2 Results for LEM dataset

5.3.2.1 Quantitative results

Best CNN-CRF variant compared with the baseline approach: As in the Campo Verde dataset, the combination of multi-loss learning - impossible transition penalization - possible transition learning, i. e. the *MCNN-CRF$_P$*

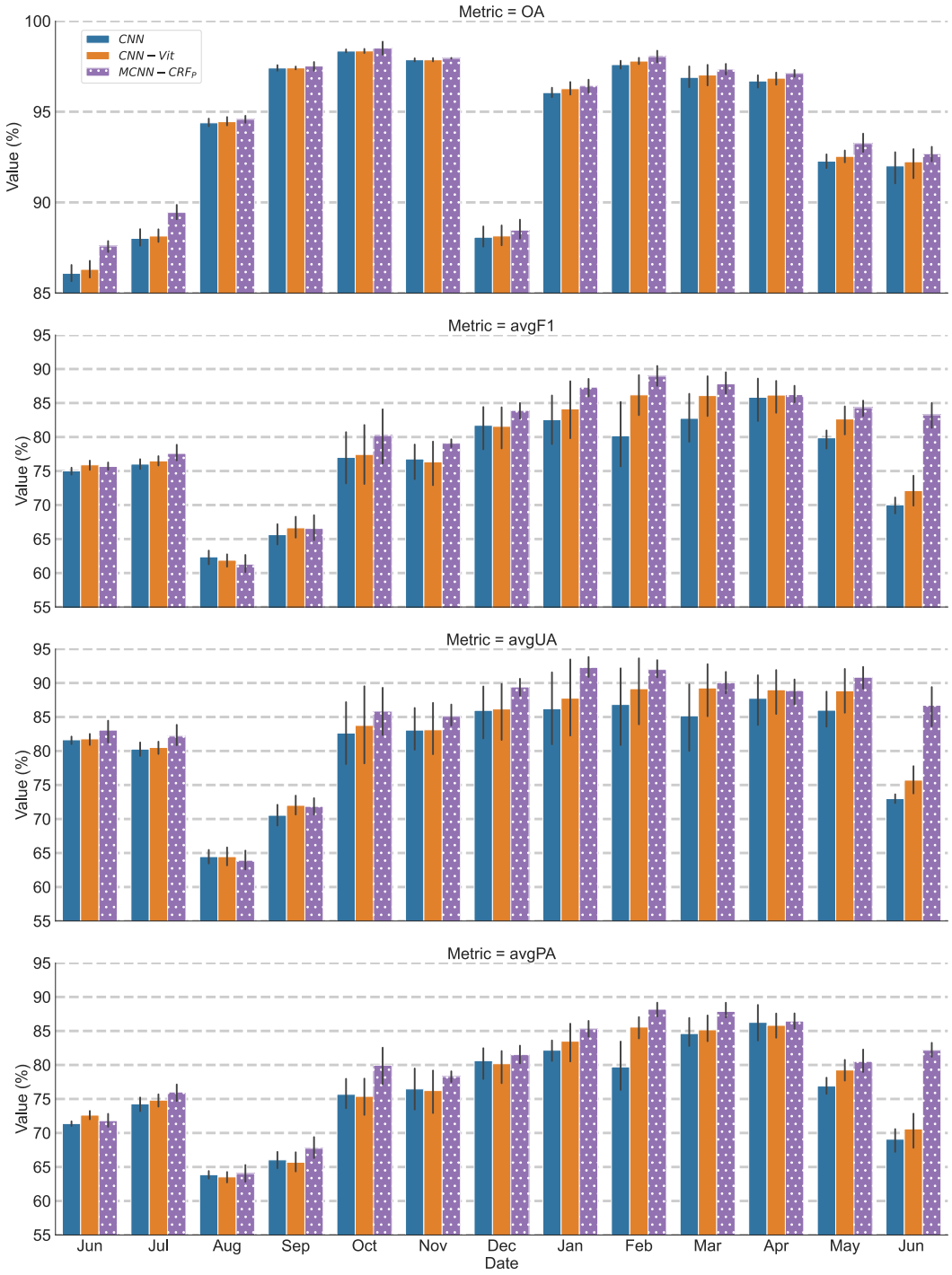

Figure 5.15: Overall Accuracy (OA), average F1 score (avgF1), average producer's accuracy (avgPA), and average user's accuracy (avgUA), computed each month (from June 2017 to June 2018) for LEM dataset for the *MCNN-CRF$_P$* model and the baseline models. Values are the average over five runs, with the black line indicating each model's minimum and maximum value.

variant, delivered the best results among all variants and the baseline approach for almost all months. Figure 5.15 summarizes the per-month results obtained for *MCNN-CRF$_P$* in terms of OA, average F1 score (avgF1), average producer's

accuracy (avgPA) and average user's accuracy (avgUA). The horizontal axis contains the month being classified. In this figure, we report just one result per month, considering the 13 annotated months for the LEM dataset. In addition, the figure presents the results for the baseline model where we report the performance for the CNN output trained solely with a per-month categorical cross-entropy (first bar in the figure) and the performance after applying the Viterbi decoding (*CNN-Vit* in the figure). Given the complex nature of class *not identified*, and the high oscillation in performance observed for all methods, we excluded this class from the classification report.

Similar to the results for Campo Verde, the Viterbi decoding (*CNN-Vit*) improved the CNN output according to all metrics in almost all months. Again, definitive improvements were observed for *MCNN-CRF$_P$* compared to the baseline method *CNN-Vit*, as well as more robust results to different network initialization, reporting slight performance variations among the five runs for most months. Looking at the OA, moderate gains were also observed for *MCNN-CRF$_P$*, achieving up to 1.3% for June 2017 and July 2017 (first and second bar groups).

Figure 5.16 reports the F1 per class brought by both models for the months with the higher differences between the baseline and the proposed model. In the figure, the horizontal axis contains the crop being classified, whereas the vertical axis contains the F1 score improvements/drops in percent for *CNN-Vit* and *MCNN-CRF$_P$* with respect to the *CNN* model. Here we observed the most significant gains in performance for class *millet* with 82%, followed by classes *conversion area* (up to 29.8%); *pasture* (31%); *hay* (up to 15%); *soybean* (14%); *maize* (7%) and *sorghum* (7%). However, we also observed significant losses for some classes on specific dates, for example, for class *hay* where *MCNN-CRF$_P$* reported a drop of 12% and 7% for months August and October, respectively, due principally to a low value in the user's accuracy. Similar behavior was observed for class *coffee* in August.

To give some insides about these variations on the F1 score, we also presented some examples of training sequences in Table 5.5. One can note that class *hay* presents high displacement in time, i.e., the start of the crop cycle varies among the different fields, and as we discussed, the model could easily take the wrong path. In this case, the temporal dynamic for rotation *hay-hay* passing for uncultivated soil can present several possible sequences, and since there are 10 times more samples with class *hay* in August than class *soil*, the possible transitions learned by *MCNN-CRF$_P$* will assign higher score to the transition *hay-hay* between July and August, than the score assigned for transition *hay-soil* for the same date.

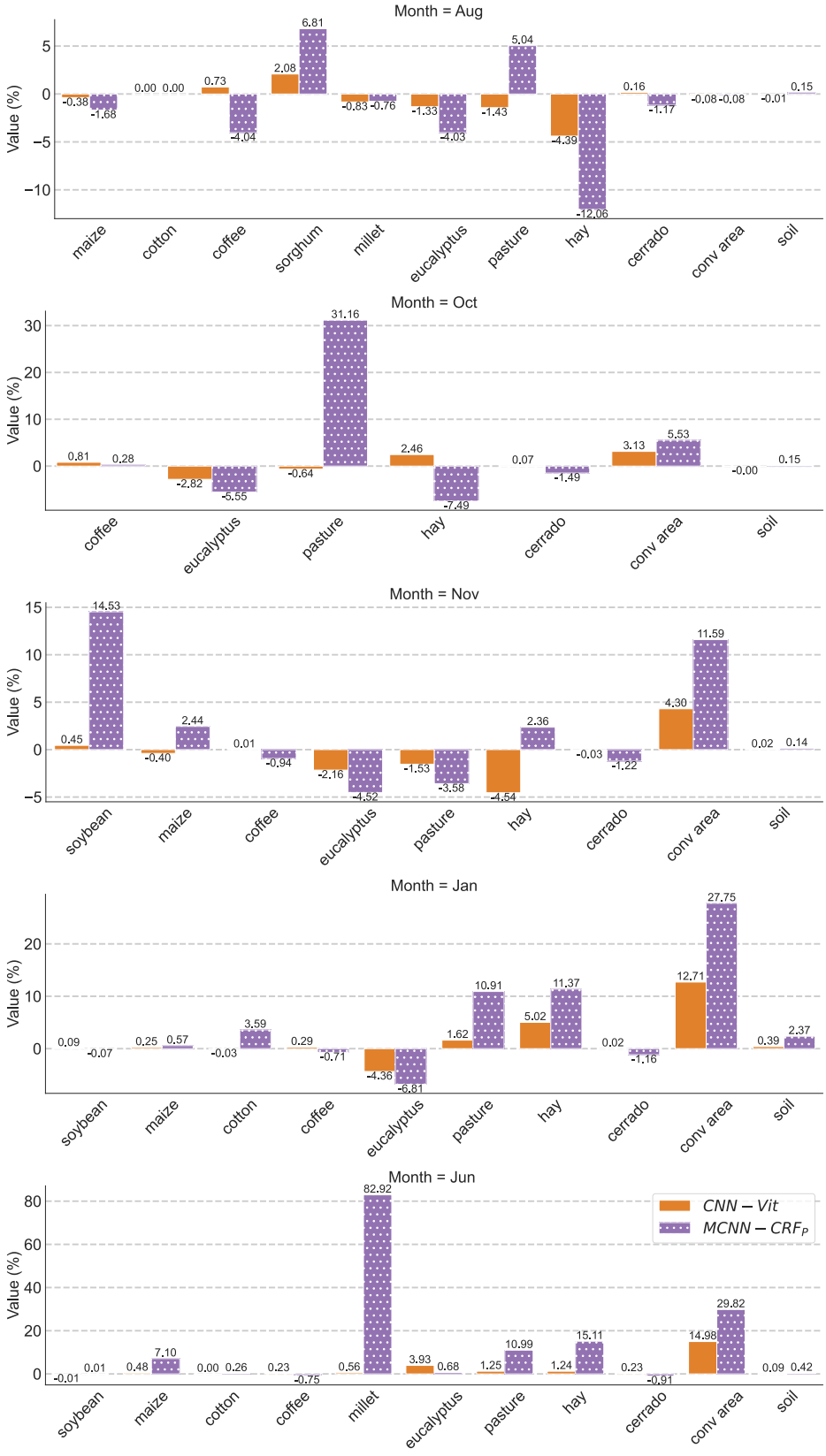

Figure 5.16: F1 score improvements/drops for *CNN-Vit* and *MCNN-CRF$_P$* with respect to *CNN*, for month August 2017, October 2017, November 2017, January 1028, and June 2018 (from top to bottom). LEM dataset.

Table 5.5: Examples of training sequences for LEM dataset. Pixl. stands for the number of training pixels for each sequence.

Pixl.	Jun	Jul	Aug	Sep	Oct	Nov	Dec
16K	*hay*	*hay*	*hay*	*soil*	*soil*	*hay*	*hay*
0.16K	*hay*	*hay*	*soil*	*soil*	*soil*	*hay*	*hay*
4K	*hay*	*hay*	*soil*	*soil*	*soil*	*soil*	*soil*

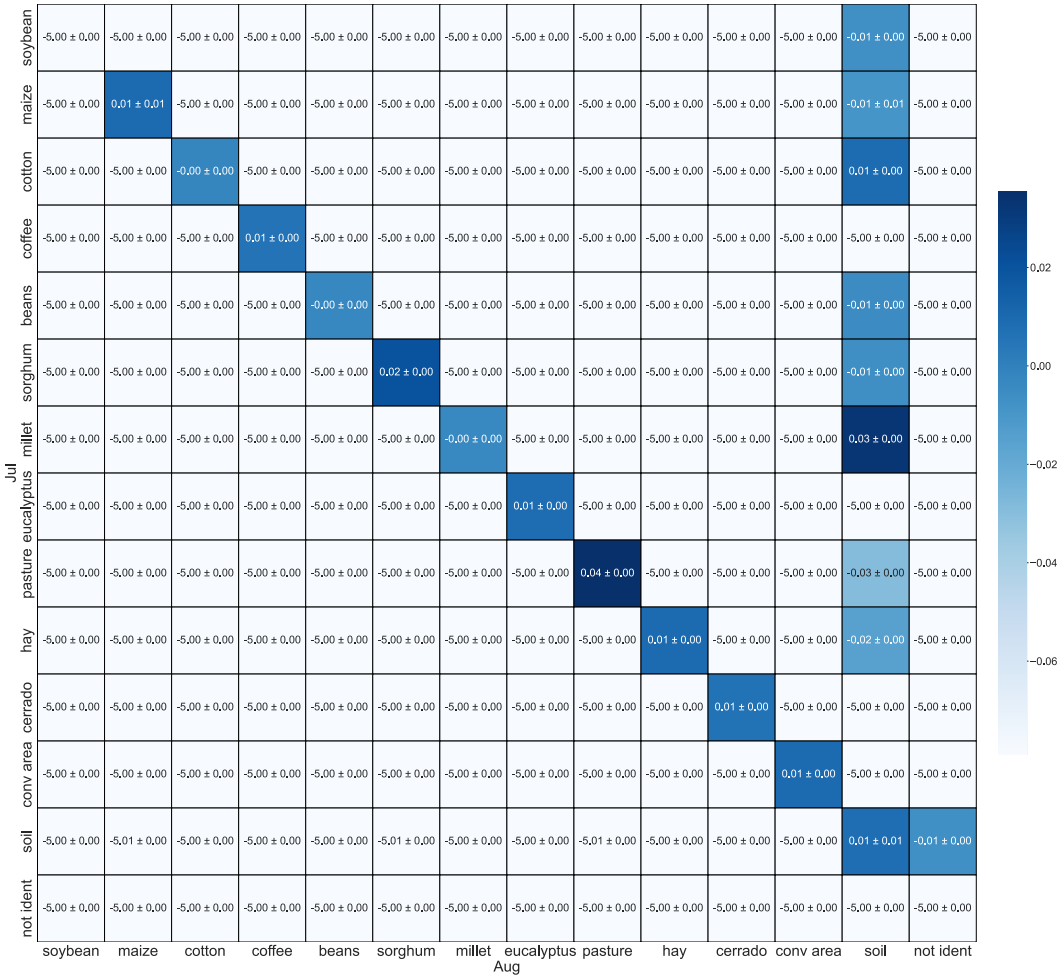

Figure 5.17: Transition matrix for adjacent months July-December learned by $MCNN\text{-}CRF_P$ model on LEM dataset.

The learned transition matrix between July and December is presented in Figure 5.17 and we can confirm the difference in score between the above-mentioned transitions, being 0.01 for *hay-hay* and -0.2 for *hay-soil*. In addition, Figure 5.18 reports the confusion matrix for August (the month with the higher drop in performance for class *hay*), and as observed, 2% of *soil* samples where miss-classified as *hay*, other minor classes as *cotton* and **conv area** were totally miss-classified.

Comparison between data-driven and prior-knowledge: Figure 5.19 summarizes the per-month results obtained for the proposed variants for schema 1

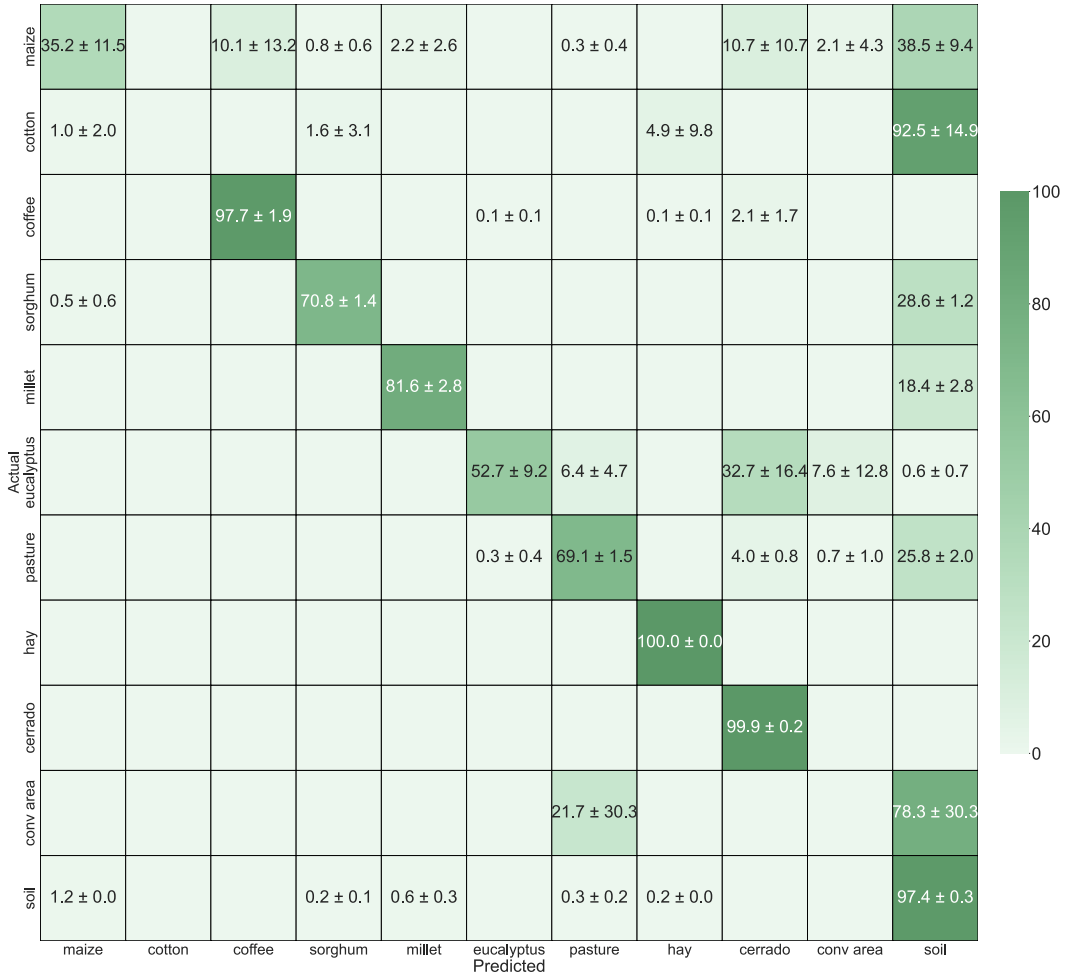

Figure 5.18: Confusion matrix for August for *MCNN-CRF$_P$* model on LEM dataset.

and 2: fixed transitions based on prior knowledge *CNN-CRF$_F$*; penalizing only the less probable transitions *CNN-CRF$_P$*; learned transitions *CNN-CRF$_L$*. We also report the results for the baseline model.

Considering the variants that employ prior knowledge, we observed that *CNN-CRF$_F$* gains to *CNN-Vit* in five out of the 13 months in terms of avgF1, reporting a drop in performance in six months, and almost equal performance for the remaining two months. Analyzing *CNN-CRF$_P$* variant, contrarily to the observed for Campo Verde, the model delivered the best results compared to *CNN-CRF$_F$* in all months. However, the error bars in the figure also indicate less robustness for *CNN-CRF$_P$* model, presenting a higher standard deviation than *CNN-CRF$_F$* for almost all months. Comparing with the variant that learns the transitions scores, the results revealed that *CNN-CRF$_P$* also outperformed *CNN-CRF$_L$* in terms of avgF1, avgPA, and avgUA, for nearly all tested months. *CNN-CRF$_L$* was generally the model with the lowest performance. These results indicate that, different from the Campo Verde dataset, learning

the transition matrices did not improve the results compared to the variant that employs prior knowledge, including the baseline approach *CNN-Vit*. It is worth pointing out that the LEM dataset has 200 unique training sequences, and when compared with the Campo Verde dataset, which only has 74 unique sequences, this is a much more complex problem. In addition, the network needs to learn a considerable number of parameters for the transition matrix (i.e., $14 \times 14 \times 12$) that must be capable of modeling all possible sequences.

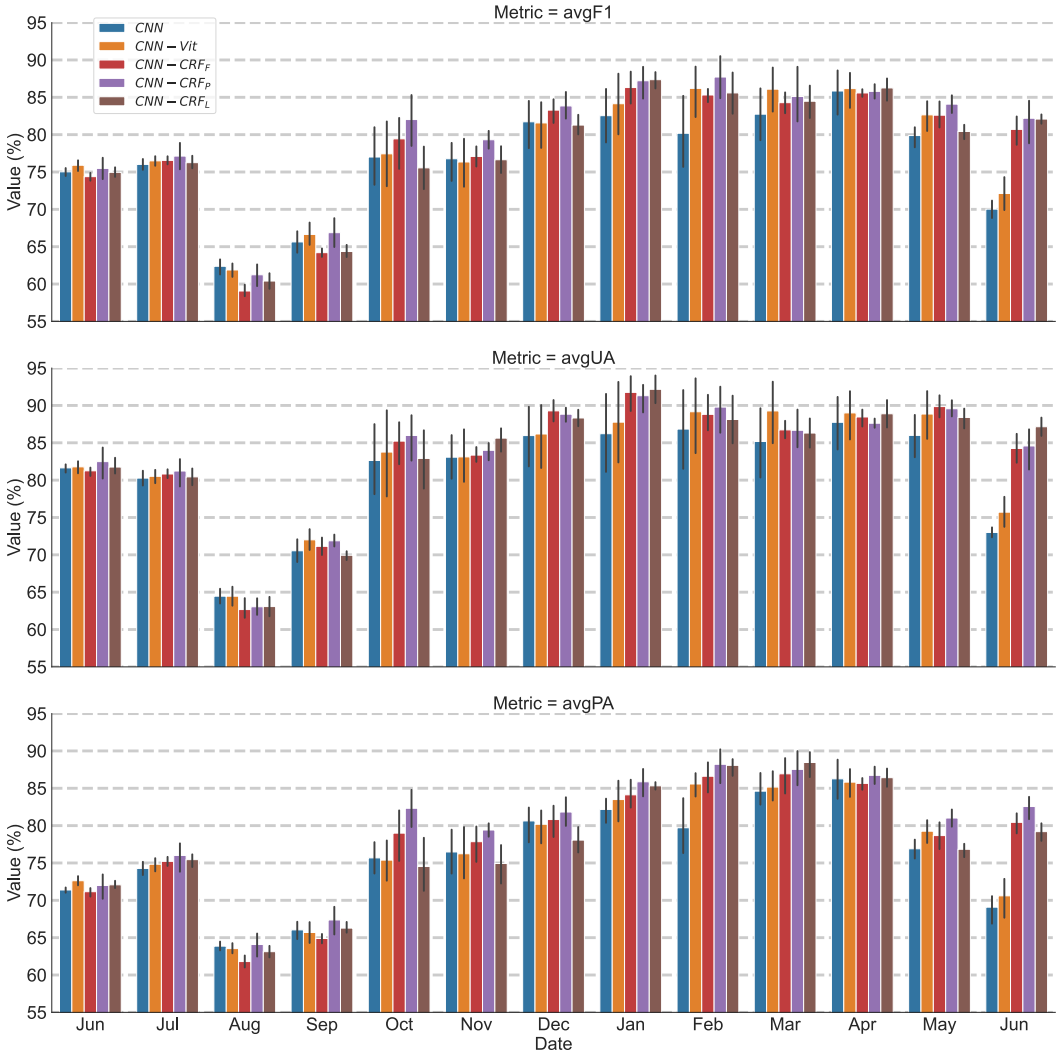

Figure 5.19: Average F1 score (avgF1), average producer's accuracy (avgPA), and average user's accuracy (avgUA), computed each month for LEM dataset for single-loss data-driven and prior-knowledge variants. Values are the average over five runs, with the black line indicating each model's minimum and maximum value.

Multi-loss learning: Figure 5.20 summarize the per-month results obtained for the proposed CNN-CRF variants now trained using the multi-loss schema. Comparing the single-loss results for the variants that use prior knowl-

edge (*CNN-CRF$_F$* and *CNN-CRF$_P$*), the corresponding multi-loss results (*MCNN-CRF$_F$* and *MCNN-CRF$_P$*), as in Campo Verde we observed that the inclusion of the per-month cross-entropy loss in our CNN-CRF models brought accuracy gains for nearly all month. It is worth noticing that *MCNN-CRF$_P$* reported a loss in performance for October due to classes *pasture* and *conversion area* that achieved discrete gains compared to the gains observed for the single-loss *CNN-CRF$_P$* variant. For the *CNN-CRF$_L$* variant, the inclusion of the cross-entropy loss brought both incremental and detrimental results or remained the same over the 13 months.

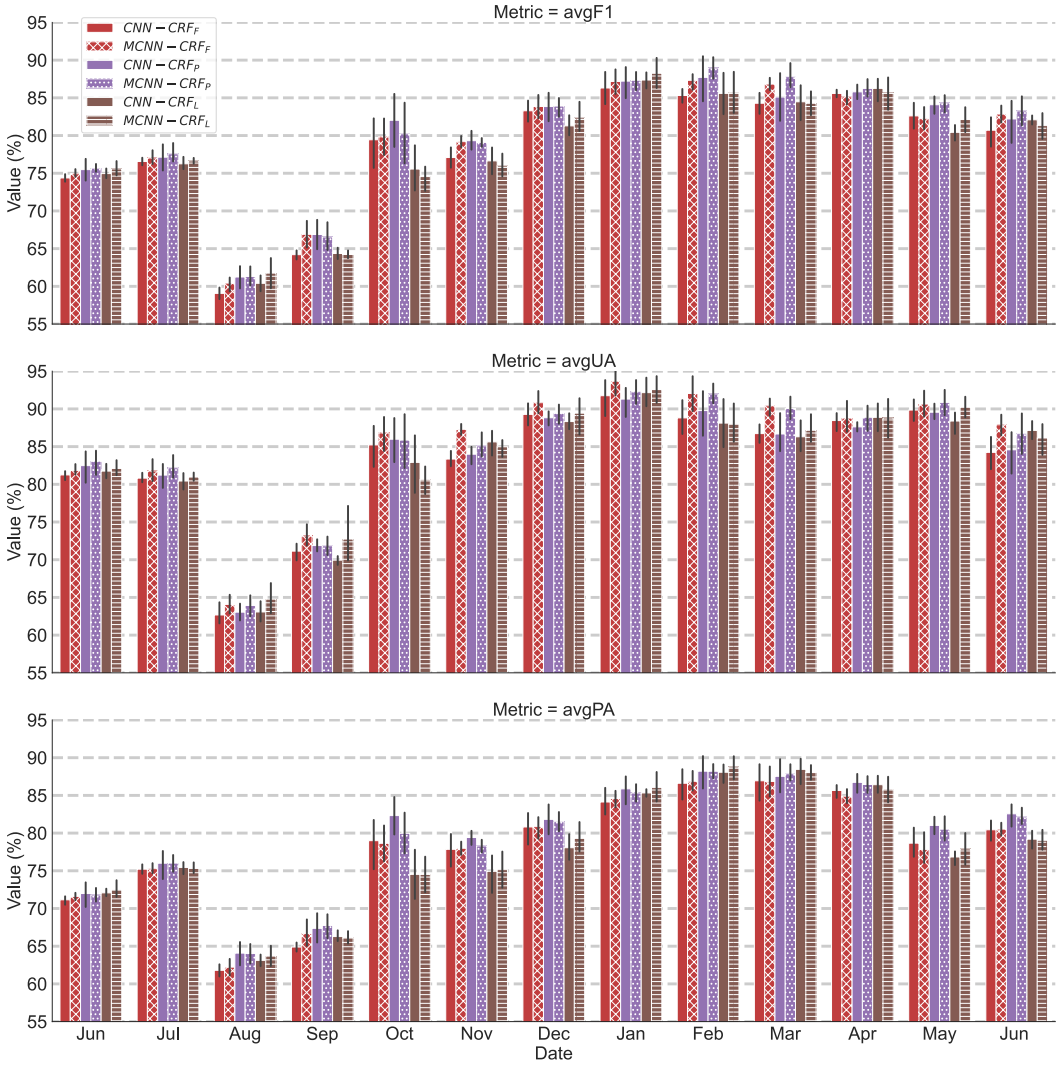

Figure 5.20: Average F1 score (avgF1), average producer's accuracy (avgPA) and average user's accuracy (avgUA), computed each month for LEM dataset for single-loss and multi-loss variants. Values are the average over five runs, with the black line indicating each model's minimum and maximum value.

Figure 5.21: Prediction and error maps for each method for selected months for LEM dataset. GT stands for ground truth. The prediction maps use the same color legend as in Figure 5.5. For the error maps, dark orange is the misclassified area.

5.3.2.2 Qualitative results

Figure 5.21 presents the classification maps for each method in a selected area. For conciseness, we showed the results from October 2017 to February 2018. In addition, Figure 5.21 reports the error maps, where the dark orange

color represents the misclassified regions. As in the Campo Verde dataset, classification errors were observed between class *soil* and other classes such as *pasture* (dark orange rectangle in the error map for October) and *soybean* (dark orange rectangle in the error map for January). The last one can be explained again due to the displacement in time during seeding time. Despite not considering the class *non identified* in our quantitative report, we can observe in the prediction maps that all models reported misclassification errors for this class.

Analyzing the entropy in emission scores (Figure 5.22), we observed the same tendency as in the Campo Verde dataset, the entropy in the prediction decreases for both *CNN-CRF$_F$* and *CNN-CRF$_P$* models when using the multi-loss training scheme.

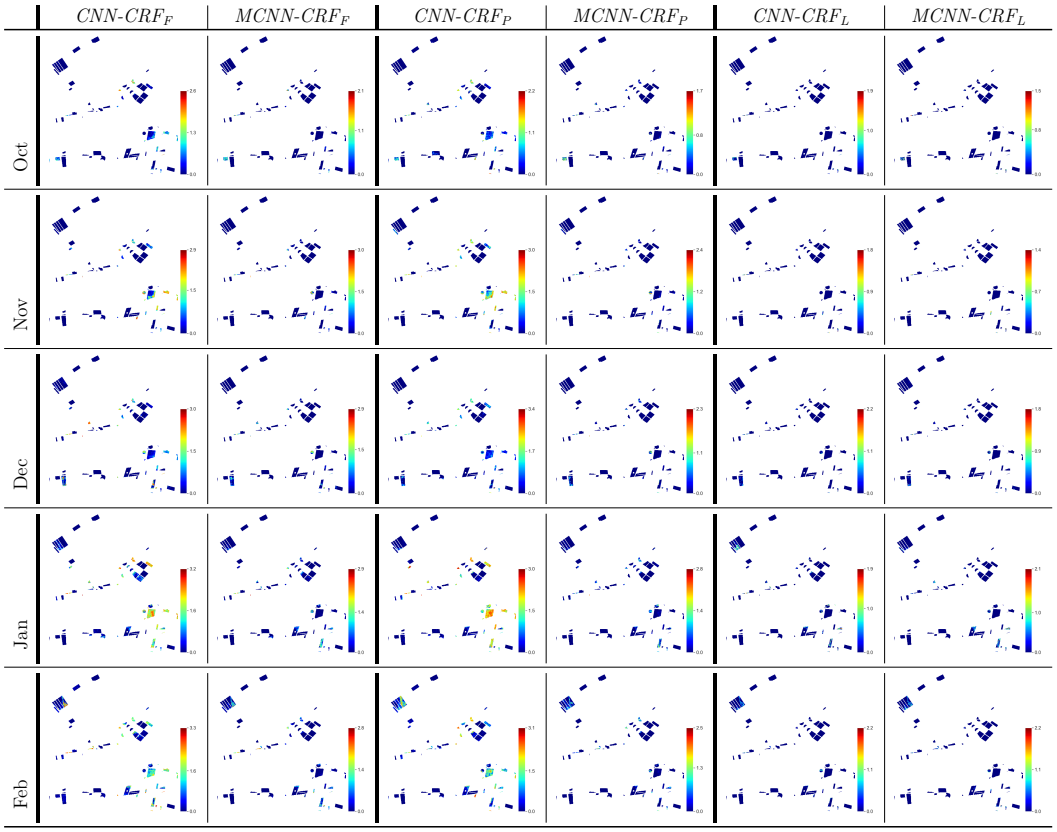

Figure 5.22: Maps of the entropy for the emission scores for CRF-based methods for a selected area from October to February for LEM dataset.

5.3.3 Experiments main conclusions

These results matched our research hypotheses. Including the CRF-loss function improved the performance compared to a CNN trained only with a per-date cross-entropy loss. In the absence of prior information about crop transitions between adjacent months, learning the transition scores from the

training data (i.e., $CNN\text{-}CRF_L$ model) improved the CNN baseline model that doesn't consider label dependency. Nonetheless, as the number of possible label sequences and the number of classes increased, using prior knowledge about possible or impossible crop transitions delivered the best results for both non-hybrid and hybrid models.

In addition, our experiments indicated that only penalizing the less probable crop transitions allowed the model to gain more flexibility when considering a large number of possible crop sequences as in the LEM dataset. Moreover, for the single-loss training setting that imposes temporal constraints based on expert knowledge, the results revealed that the models tended to be less confident in the emission scores (i.e., high entropy).

Adding a second loss function that focuses on improving the per-date accuracy delivered the best performance for those models that consider the constraints about the less probable transitions. As expected, training only with the CRF-loss forces the network to find the set of parameters that maximize the sequence labeling, whereas using the second loss function increased the model performance also for each independent month. The multi-loss training setting decreased the entropy in the emission scores, and this low entropy indicates that the models had higher confidence in their predictions.

Finally, Figure 5.23 presents the OA at the sequence level for both datasets for each model. As observed, the models that consider the temporal label dependencies consistently outperformed the baseline CNN model. Moreover, the graphics show that the multi-loss schema can sometimes be detrimental. However, as discussed above, since the second loss function aims to improve the per-date accuracy (favoring less frequent classes and sequences), this slight drop in performance at the sequence level is expected, principally for datasets like LEM with a high number of different label sequences.

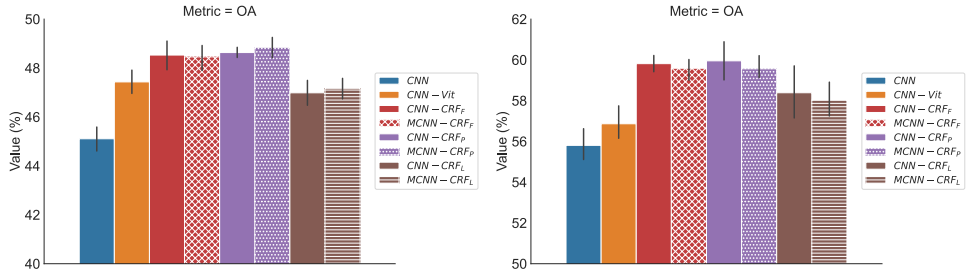

Figure 5.23: Overall Accuracy at sequence level for Campo Verde (left) and LEM datasets (right).

6
Conclusions and future steps

This work introduced a hybrid deep learning architecture for multi-temporal crop recognition, which combines the spatio-temporal context encoding of convolutional networks and the temporal modeling capabilities of conditional random fields (CRF) in an end-to-end framework. The proposed end-to-end framework consists of three modules: a 3D fully convolutional network (FCN), a linear-chain CRF module, and a Viterbi algorithm that delivers the final sequence for each pixel at inference time. The FCN learns spatio-temporal features and provides the emission that serves as input to the CRF module; the transition scores are also part of the CRF module. The training of FCN and CRF modules leans on an end-to-end learning procedure using a CRF-based loss function.

Unlike similar approaches that learn a single global transition matrix that models temporal dynamics, our method learns a different transition matrix for each pair of adjacent dates from training data. Furthermore, the method enforces prior knowledge about the temporal dynamics of cultures in the learning process. The prior knowledge consists of information about possible and less probable crop transitions within a target region. In addition, we proposed a multi-loss training scheme that forces the network to learn the set of parameters that maximize both the sequence labeling and the per-date accuracy.

We tested the models upon two publicly available multi-temporal SAR datasets from two tropical regions in Brazil with highly complex Spatio-temporal crop dynamics. The experiments indicated that the proposed end-to-end frameworks outperformed the baseline model that employs feature learning and temporal modeling in two separate stages. The improvements were particularly apparent in the F1 score values, where the hybrid models generally reported higher user's and producer's accuracies.

The experimental analysis demonstrated the potential of learning temporal crop dynamics in tropical areas from the training data. The results indicated that learning from the data is a feasible option, reporting high accuracy values; however, as the temporal complexity increases, we observed a drop in the model's generalization capability. Increased complexity demands more an-

notated samples to properly reflect the temporal dynamics in the target field. In contrast, including priors about the temporal crop dynamics delivered the best results when considering a high number of different crop rotations (i.e., label sequences) in the target regions.

Although generally successful in predicting the correct crop type at each date, the model with prior knowledge trained solely with the CRF-based loss delivered emission scores with high entropy due to the constraint imposed for less probable transitions. Considering this, adding a second loss function that focuses on improving the per-date accuracy in combination with prior knowledge about temporal dynamics generally delivered the best performance for both datasets, improving the network's confidence in the prediction and the per-date accuracy.

Finally, it is worth mentioning that the level of flexibility of the imposed temporal constraint greatly impacted the classification performance. In a future step, we plan to perform a sensitivity analysis of this hyperparameter. We also intend to explore a higher order CRF that considers class dependencies between non-adjacent epochs.

Bibliography

1 ACHANCCARAY DIAZ, 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, Doctoral dissertation, PUC-Rio, 2019.

2 SANCHES, I.; LUIZ, A. J. B.; MONTIBELLER, B.; SCHULTZ, B.; TRABAQUINI, K.; EBERHARDT, D.; FORMAGGIO, A. ; MAURANO, L.. **Understanding the dynamic of tropical agriculture for remote sensing applications: a case study of southeastern brazil.** Embrapa Meio Ambiente-Artigo em periódico indexado (ALICE), 2019.

3 SUTTON, C.; MCCALLUM, A.. **An introduction to conditional random fields for relational learning**. Introduction to statistical relational learning, 2:93–128, 2006.

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

5 MARTINEZ, J. A. C.; LA ROSA, L. E. C.; FEITOSA, R. Q.; SANCHES, I. D. ; HAPP, P. N.. **Fully convolutional recurrent networks for multidate crop recognition from multitemporal image sequences**. ISPRS Journal of Photogrammetry and Remote Sensing, 171:188–201, 2021.

6 FAO, F.; OTHERS. **The future of food and agriculture–trends and challenges**. Annual Report, 296, 2017.

7 NATIONS, U.. **World population prospects 2019: Highlights.** Department of Economic and Social Affairs, Population Division, ST/ESA/SER.A/423, 2019.

8 RAMANKUTTY, N.; MEHRABI, Z.; WAHA, K.; JARVIS, L.; KREMEN, C.; HERRERO, M. ; RIESEBERG, L. H.. **Trends in global agricultural land use: implications for environmental health and food security**. Annual review of plant biology, 69:789–815, 2018.

9 BODIRSKY, B. L.; DIETRICH, J. P.; MARTINELLI, E.; STENSTAD, A.; PRADHAN, P.; GABRYSCH, S.; MISHRA, A.; WEINDL, I.; LE MOUËL, C.; ROLINSKI, S. ; OTHERS. **The ongoing nutrition transition thwarts long-term targets for food security, public health and environmental protection**. Scientific reports, 10(1):1–14, 2020.

10 FAO. **World Food and Agriculture - Statistical Yearbook**. Rome, 2020.

11 ANDERSON, J. R.. **A land use and land cover classification system for use with remote sensor data**, volumen 964. US Government Printing Office, 1976.

12 MORAN, M. S.; INOUE, Y. ; BARNES, E.. **Opportunities and limitations for image-based remote sensing in precision crop management**. Remote sensing of Environment, 61(3):319–346, 1997.

13 PANIGRAHY, S.; SHARMA, S.. **Mapping of crop rotation using multidate indian remote sensing satellite digital data**. ISPRS Journal of Photogrammetry and Remote Sensing, 52(2):85–91, 1997.

14 WARDLOW, B. D.; EGBERT, S. L.. **Large-area crop mapping using time-series modis 250 m ndvi data: An assessment for the u.s. central great plains**. volumen 112, p. 1096 – 1116. 2008.

15 IMMITZER, M.; VUOLO, F. ; ATZBERGER, C.. **First experience with sentinel-2 data for crop and tree species classifications in central europe**. Remote Sensing, 8(3):166, 2016.

16 LU, D.; WENG, Q.. **A survey of image classification methods and techniques for improving classification performance**. International journal of Remote sensing, 28(5):823–870, 2007.

17 BELGIU, M.; CSILLIK, O.. **Sentinel-2 cropland mapping using pixel-based and object-based time-weighted dynamic time warping analysis**. Remote sensing of environment, 204:509–523, 2018.

18 MATTON, N.; CANTO, G. S.; WALDNER, F.; VALERO, S.; MORIN, D.; INGLADA, J.; ARIAS, M.; BONTEMPS, S.; KOETZ, B. ; DEFOURNY, P.. **An automated method for annual cropland mapping along the season for various globally-distributed agrosystems using high spatial and temporal resolution time series**. Remote Sensing, 7(10):13208–13232, 2015.

19 SKAKUN, S.; KUSSUL, N.; SHELESTOV, A. Y.; LAVRENIUK, M. ; KUSSUL, O.. **Efficiency Assessment of Multitemporal C-Band Radarsat-2 Intensity and Landsat-8 Surface Reflectance Satellite Imagery for Crop Classification in Ukraine**. IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing, 9(8):3712–3719, 2016.

20 KARTHIKEYAN, L.; CHAWLA, I. ; MISHRA, A. K.. **A review of remote sensing applications in agriculture for food security: Crop growth and yield, irrigation, and crop losses**. Journal of Hydrology, 586:124905, 2020.

21 PRUDENTE, V. H. R.; MARTINS, V. S.; VIEIRA, D. C.; E SILVA, N. R. D. F.; ADAMI, M. ; SANCHES, I. D.. **Limitations of cloud cover for optical remote sensing of agricultural areas across south america**. Remote Sensing Applications: Society and Environment, 20:100414, 2020.

22 ATKINSON, P. M.; TATNALL, A.. **Introduction neural networks in remote sensing**. International Journal of remote sensing, 18(4):699–709, 1997.

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

24 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.

25 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, p. 7–9, 2012.

26 INGLADA, J.; ARIAS, M.; TARDY, B.; MORIN, D.; VALERO, S.; HAGOLLE, O.; DEDIEU, G.; SEPULCRE, G.; BONTEMPS, S. ; DEFOURNY, P.. **Benchmarking of algorithms for crop type land-cover maps using Sentinel-2 image time series**. International Geoscience and Remote Sensing Symposium (IGARSS), 2015-Novem:3993–3996, 2015.

27 WALDNER, F.; CANTO, G. S. ; DEFOURNY, P.. **Automated annual cropland mapping using knowledge-based temporal features**. ISPRS Journal of Photogrammetry and Remote Sensing, 110:1–13, 2015.

28 LUCIEER, A.; STEIN, A. ; FISHER, P.. **Multivariate texture-based segmentation of remotely sensed imagery for extraction of objects and their uncertainty**. International Journal of Remote Sensing, 26(14):2917–2936, 2005.

29 RUIZ, L.; FDEZ-SARRÍA, A. ; RECIO, J.. **Texture feature extraction for classification of remote sensing data using wavelet decomposition: a comparative study**. In: 20TH ISPRS CONGRESS, volumen 35, p. 1109–1114, 2004.

30 HE, D.-C.; WANG, L.. **Texture unit, texture spectrum, and texture analysis**. IEEE transactions on Geoscience and Remote Sensing, 28(4):509–512, 1990.

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

32 PEÑA-BARRAGÁN, J. M.; NGUGI, M. K.; PLANT, R. E. ; SIX, J.. **Object-based crop identification using multiple vegetation indices, textural features and crop phenology**. Remote Sensing of Environment, 115(6):1301–1316, 2011.

33 TANG, Z.; WANG, H.; LI, X.; LI, X.; CAI, W. ; HAN, C.. **An object-based approach for mapping crop coverage using multiscale weighted and machine learning methods**. IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing, 13:1700–1713, 2020.

34 SON, N.-T.; CHEN, C.-F.; CHEN, C.-R.; TOSCANO, P.; CHENG, Y.-S.; GUO, H.-Y. ; SYU, C.-H.. **A phenological object-based approach for rice crop classification using time-series sentinel-1 synthetic aperture radar (sar) data in taiwan**. International Journal of Remote Sensing, 42(7):2722–2739, 2021.

35 MELGANI, F.; SERPICO, S. B.. **A markov random field approach to spatio-temporal contextual image classification**. IEEE Transactions on Geoscience and Remote Sensing, 41(11):2478–2487, 2003.

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

37 SIACHALOU, S.; MALLINIS, G. ; TSAKIRI-STRATI, M.. **A hidden markov models approach for crop classification: Linking crop**

phenology to time series of multi-sensor remote sensing data. Remote Sensing, 7(4):3633–3650, 2015.

38 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.

39 KENDUIYWO, B. K.; BARGIEL, D. ; SOERGEL, U.. **Higher order dynamic conditional random fields ensemble for crop type classification in radar images**. IEEE Transactions on Geoscience and Remote Sensing, 2017.

40 HOBERG, T.; ROTTENSTEINER, F. ; HEIPKE, C.. **Classification of multitemporal remote sensing data using Conditional Random Fields**. In: 2010 IAPR WORKSHOP ON PATTERN RECOGNITION IN REMOTE SENSING, p. 1–4. IEEE, aug 2010.

41 CASTELLAZZI, M.; WOOD, G.; BURGESS, P. J.; MORRIS, J.; CONRAD, K. ; PERRY, J.. **A systematic representation of crop rotations**. Agricultural Systems, 97(1-2):26–33, 2008.

42 FIRAT, O.; CAN, G. ; VURAL, F. T. Y.. **Representation learning for contextual object and region detection in remote sensing**. In: PATTERN RECOGNITION (ICPR), 2014 22ND INTERNATIONAL CONFERENCE ON, p. 3708–3713. IEEE, 2014.

43 ROMERO, A.; GATTA, C. ; CAMPS-VALLS, G.. **Unsupervised deep feature extraction for remote sensing image classification**. IEEE Transactions on Geoscience and Remote Sensing, 54(3):1349–1362, 2016.

44 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.

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

46 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):1217, 2018.

47 ZHONG, L.; HU, L. ; ZHOU, H.. **Deep learning based multi-temporal crop classification**. Remote Sensing of Environment, 221:430–443, 2019.

48 LECUN, Y.; BOSER, B.; DENKER, J. S.; HENDERSON, D.; HOWARD, R. E.; HUBBARD, W. ; JACKEL, L. D.. **Backpropagation applied to handwritten zip code recognition**. Neural computation, 1(4):541–551, 1989.

49 LECUN, Y.; BOTTOU, L.; BENGIO, Y. ; HAFFNER, P.. **Gradient-based learning applied to document recognition**. Proceedings of the IEEE, 86(11):2278–2324, 1998.

50 CASTRO, J. D. B.; FEITOZA, R. Q.; ROSA, L. C. L.; DIAZ, P. M. A. ; SANCHES, I. D. A.. **A Comparative Analysis of Deep Learning Techniques for Sub-Tropical Crop Types Recognition from Multitemporal Optical/SAR Image Sequences**. 2017 30th SIBGRAPI Conference on Graphics, Patterns and Images (SIBGRAPI), p. 382–389, 2017.

51 JI, S.; ZHANG, C.; XU, A.; SHI, Y. ; DUAN, Y.. **3d convolutional neural networks for crop classification with multi-temporal remote sensing images**. Remote Sensing, 10(1):75, 2018.

52 ROGOZINSKI, M.; MARTINEZ, J. A. C. ; FEITOSA, R. Q.. **3d convolution for multidate crop recognition from multitemporal image sequences**. International Journal of Remote Sensing, p. 1–23, 2021.

53 CUÉ LA ROSA, L. E.; QUEIROZ FEITOSA, R.; NIGRI HAPP, P.; DEL'ARCO SANCHES, I. ; OSTWALD PEDRO DA COSTA, G. A.. **Combining deep learning and prior knowledge for crop mapping in tropical regions from multitemporal sar image sequences**. Remote Sensing, 11(17):2029, 2019.

54 LIU, C.; SONG, W.; LU, C. ; XIA, J.. **Spatial-temporal hidden markov model for land cover classification using multitemporal satellite images**. IEEE Access, 9:76493–76502, 2021.

55 MA, X.; HOVY, E.. **End-to-end sequence labeling via bi-directional lstm-cnns-crf**. arXiv preprint arXiv:1603.01354, 2016.

56 LIU, L.; SHANG, J.; REN, X.; XU, F.; GUI, H.; PENG, J. ; HAN, J.. **Empower sequence labeling with task-aware neural language model**. In: PROCEEDINGS OF THE AAAI CONFERENCE ON ARTIFICIAL INTELLIGENCE, volumen 32, 2018.

57 LI, J.; TODOROVIC, S.. **Set-constrained viterbi for set-supervised action segmentation**. In: PROCEEDINGS OF THE IEEE/CVF CONFERENCE ON COMPUTER VISION AND PATTERN RECOGNITION, p. 10820–10829, 2020.

58 RYERSON, R.; HENDERSON, F.; LEWIS, A.; FOR PHOTOGRAMMETRY, A. S. ; SENSING, R.. **Manual of Remote Sensing, Principles and Applications of Imaging Radar**. Manual of Remote Sensing - Third Edition. Wiley, 1998.

59 CURLANDER, J.; MCDONOUGH, R.. **Synthetic Aperture Radar: Systems and Signal Processing**. Wiley Series in Remote Sensing and Image Processing. Wiley, 1991.

60 HENDERSON, F. M.; CHASAN, R.; PORTOLESE, J. ; HART JR, T.. **Evaluation of sar-optical imagery synthesis techniques in a complex coastal ecosystem**. Photogrammetric Engineering and Remote Sensing, 68(8):839–846, 2002.

61 FORKUOR, G.; CONRAD, C.; THIEL, M.; ULLMANN, T. ; ZOUNGRANA, E.. **Integration of optical and synthetic aperture radar imagery for improving crop mapping in northwestern benin, west africa**. Remote Sensing, 6(7):6472–6499, 2014.

62 MOREIRA, A.; PRATS-IRAOLA, P.; YOUNIS, M.; KRIEGER, G.; HAJNSEK, I. ; PAPATHANASSIOU, K. P.. **A tutorial on synthetic aperture radar**. IEEE Geoscience and remote sensing magazine, 1(1):6–43, 2013.

63 HAACK, B.. **A comparison of land use/cover mapping with varied radar incident angles and seasons**. GIScience & Remote Sensing, 44(4):305–319, 2007.

64 SORIA-RUIZ, J.; FERNANDEZ-ORDONEZ, Y. ; MCNAIRN, H.. **Corn monitoring and crop yield using optical and microwave remote sensing**. In: GEOSCIENCE AND REMOTE SENSING. IntechOpen, 2009.

65 JIA, K.; LI, Q.; TIAN, Y.; WU, B.; ZHANG, F. ; MENG, J.. **Crop classification using multi-configuration sar data in the north china plain**. International Journal of Remote Sensing, 33(1):170–183, 2012.

66 ULABY, F.; BARE, J.. **Look direction modulation function of the radar backscattering coefficient of agricultural fields.** Photogrammetric Engineering and Remote Sensing, 45(11):1495–1506, 1 1979.

67 BRISCO, B.; BROWN, R. J.; SNIDER, B.; SOFKO, G. J.; KOEHLER, J. A. ; WACKER, A. G.. **Tillage effects on the radar backscattering coefficient of grain stubble fields**. International Journal of Remote Sensing, 12(11):2283–2298, 1991.

68 BRISCO, B.; BROWN, R.; GAIRNS, J. ; SNIDER, B.. **Temporal ground-based scatterometer observations of crops in western canada**. Canadian Journal of Remote Sensing, 18(1):14–21, 1992.

69 OSMAN, J.; INGLADA, J. ; DEJOUX, J.-F.. **Assessment of a markov logic model of crop rotations for early crop mapping**. Computers and Electronics in Agriculture, 113:234–243, 2015.

70 SHAH, K. K.; MODI, B.; PANDEY, H. P.; SUBEDI, A.; ARYAL, G.; PANDEY, M. ; SHRESTHA, J.. **Diversified crop rotation: an approach for sustainable agriculture production**. Advances in Agriculture, 2021, 2021.

71 PIRES, G. F.; ABRAHÃO, G. M.; BRUMATTI, L. M.; OLIVEIRA, L. J.; COSTA, M. H.; LIDDICOAT, S.; KATO, E. ; LADLE, R. J.. **Increased climate risk in brazilian double cropping agriculture systems: Implications for land use in northern brazil**. Agricultural and forest meteorology, 228:286–298, 2016.

72 RABINER, L. R.. **A tutorial on hidden markov models and selected applications in speech recognition**. Proceedings of the IEEE, 77(2):257–286, 1989.

73 LEITE, P. B. C.; FEITOSA, R. Q.; FORMAGGIO, A. R.; DA COSTA, G. A. O. P.; PAKZAD, K. ; SANCHES, I. D. A.. **Hidden Markov Models for crop recognition in remote sensing image sequences**. Pattern Recognition Letters, 32(1):19–26, 2011.

74 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.

75 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.

76 ZHAO, H.; CHEN, Z.; JIANG, H.; JING, W.; SUN, L. ; FENG, M.. **Evaluation of three deep learning models for early crop classification using sentinel-1a imagery time series—a case study in zhanjiang, china**. Remote Sensing, 11(22):2673, 2019.

77 ZHAO, H.; DUAN, S.; LIU, J.; SUN, L. ; REYMONDIN, L.. **Evaluation of five deep learning models for crop type mapping using sentinel-2 time series images with missing information**. Remote Sensing, 13(14):2790, 2021.

78 XU, J.; YANG, J.; XIONG, X.; LI, H.; HUANG, J.; TING, K.; YING, Y. ; LIN, T.. **Towards interpreting multi-temporal deep learning models in crop mapping**. Remote Sensing of Environment, 264:112599, 2021.

79 XU, J.; ZHU, Y.; ZHONG, R.; LIN, Z.; XU, J.; JIANG, H.; HUANG, J.; LI, H. ; LIN, T.. **Deepcropmapping: A multi-temporal deep learning approach with improved spatial generalizability for dynamic corn and soybean mapping**. Remote Sensing of Environment, 247:111946, 2020.

80 MORENO-REVELO, M. Y.; GUACHI-GUACHI, L.; GÓMEZ-MENDOZA, J. B.; REVELO-FUELAGÁN, J. ; PELUFFO-ORDÓÑEZ, D. H.. **Enhanced convolutional-neural-network architecture for crop classification**. Applied Sciences, 11(9):4292, 2021.

81 SEYDI, S. T.; AMANI, M. ; GHORBANIAN, A.. **A dual attention convolutional neural network for crop classification using time-series sentinel-2 imagery**. Remote Sensing, 14(3):498, 2022.

82 JI, S.; ZHANG, C.; XU, A.; SHI, Y. ; DUAN, Y.. **3d convolutional neural networks for crop classification with multi-temporal remote sensing images**. Remote Sensing, 10(1), 2018.

83 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.

84 VOLPI, M.; TUIA, D.. **Dense semantic labeling of subdecimeter resolution images with convolutional neural networks**. IEEE Transactions on Geoscience and Remote Sensing, 55(2):881–893, 2017.

85 MAGGIORI, E.; TARABALKA, Y.; CHARPIAT, G. ; ALLIEZ, P.. **Convolutional neural networks for large-scale remote-sensing image classification**. IEEE Transactions on Geoscience and Remote Sensing, 55(2):645–657, 2017.

86 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.

87 WEI, P.; CHAI, D.; LIN, T.; TANG, C.; DU, M. ; HUANG, J.. **Large-scale rice mapping under different years based on time-series sentinel-1 images using deep semantic segmentation model**. ISPRS journal of photogrammetry and remote sensing, 174:198–214, 2021.

88 LA ROSA, L. C.; OLIVEIRA, D. ; FEITOSA, R.. **Investigating fusion strategies on encoder-decoder networks for crop segmentation using sar and optical image sequences**. In: 2021 IEEE INTERNATIONAL GEOSCIENCE AND REMOTE SENSING SYMPOSIUM IGARSS, p. 2405–2408. IEEE, 2021.

89 M RUSTOWICZ, R.; CHEONG, R.; WANG, L.; ERMON, S.; BURKE, M. ; LOBELL, D.. **Semantic segmentation of crop type in africa: A novel dataset and analysis of deep learning methods**. In: PROCEEDINGS OF THE IEEE/CVF CONFERENCE ON COMPUTER VISION AND PATTERN RECOGNITION WORKSHOPS, p. 75–82, 2019.

90 MOHAMMADI, S.; BELGIU, M. ; STEIN, A.. **3d fully convolutional neural networks with intersection over union loss for crop mapping from multi-temporal satellite images**. In: 2021 IEEE INTERNATIONAL GEOSCIENCE AND REMOTE SENSING SYMPOSIUM IGARSS, p. 5834–5837. IEEE, 2021.

91 WANG, J.; DU, Z.; LI, A. ; WANG, Y.. **Atrous temporal convolutional network for video action segmentation**. In: 2019 IEEE INTERNATIONAL CONFERENCE ON IMAGE PROCESSING (ICIP), p. 1585–1589. IEEE, 2019.

92 ALONSO, I.; CAMBRA, A.; MUNOZ, A.; TREIBITZ, T. ; MURILLO, A. C.. **Coral-segmentation: Training dense labeling models with sparse ground truth**. In: PROCEEDINGS OF THE IEEE INTERNATIONAL CONFERENCE ON COMPUTER VISION WORKSHOPS, p. 2874–2882, 2017.

93 MAGGIOLO, L.; MARCOS, D.; MOSER, G. ; TUIA, D.. **Improving maps from CNNs trained with sparse, scribbled ground truths using fully connected CRFs**. In: IGARSS 2018-2018 IEEE INTERNATIONAL GEOSCIENCE AND REMOTE SENSING SYMPOSIUM, p. 2099–2102. IEEE, 2018.

94 WU, W.; QI, H.; RONG, Z.; LIU, L. ; SU, H.. **Scribble-Supervised Segmentation of Aerial Building Footprints Using Adversarial Learning**. IEEE Access, 6:58898–58911, 2018.

95 TANG, M.; DJELOUAH, A.; PERAZZI, F.; BOYKOV, Y. ; SCHROERS, C.. **Normalized cut loss for weakly-supervised cnn segmentation**. In: PROCEEDINGS OF THE IEEE CONFERENCE ON COMPUTER VISION AND PATTERN RECOGNITION, p. 1818–1827, 2018.

96 NOWOZIN, S.; LAMPERT, C. H.. **Structured learning and prediction in computer vision**, volumen 6. Now publishers Inc, 2011.

97 BISHOP, C. M.. **Pattern recognition**. Machine learning, 128(9), 2006.

98 SPIEGELHALTER, D. J.. **Bayesian graphical modelling: a case-study in monitoring health outcomes**. Journal of the Royal Statistical Society: Series C (Applied Statistics), 47(1):115–133, 1998.

99 KOLLER, D.; FRIEDMAN, N.. **Probabilistic graphical models: principles and techniques**. MIT press, 2009.

100 KLINGER, R.; TOMANEK, K.. **Classical probabilistic models and conditional random fields**. Citeseer, 2007.

101 LAFFERTY, J.; MCCALLUM, A. ; PEREIRA, F. C.. **Conditional random fields: Probabilistic models for segmenting and labeling sequence data**. 2001.

102 VITERBI, A.. **Error bounds for convolutional codes and an asymptotically optimum decoding algorithm**. IEEE transactions on Information Theory, 13(2):260–269, 1967.

103 IOFFE, S.; SZEGEDY, C.. **Batch normalization: Accelerating deep network training by reducing internal covariate shift**. In: INTERNATIONAL CONFERENCE ON MACHINE LEARNING, p. 448–456. PMLR, 2015.

104 ZHU, X. X.; TUIA, D.; MOU, L.; XIA, G.-S.; ZHANG, L.; XU, F. ; FRAUNDORFER, F.. **Deep learning in remote sensing: A comprehensive review and list of resources**. IEEE Geoscience and Remote Sensing Magazine, 5(4):8–36, 2017.

105 MA, L.; LIU, Y.; ZHANG, X.; YE, Y.; YIN, G. ; JOHNSON, B. A.. **Deep learning in remote sensing applications: A meta-analysis and review**. ISPRS journal of photogrammetry and remote sensing, 152:166–177, 2019.

106 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.

107 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.

108 ZHAO, H.; SHI, J.; QI, X.; WANG, X. ; JIA, J.. **Pyramid scene parsing network**. In: PROCEEDINGS OF THE IEEE CONFERENCE ON COMPUTER VISION AND PATTERN RECOGNITION, p. 2881–2890, 2017.

109 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.

110 HE, K.; ZHANG, X.; REN, S. ; SUN, J.. **Deep residual learning for image recognition**. In: PROCEEDINGS OF THE IEEE CONFERENCE ON COMPUTER VISION AND PATTERN RECOGNITION, p. 770–778, 2016.

111 JI, S.; ZHANG, Z.; ZHANG, C.; WEI, S.; LU, M. ; DUAN, Y.. **Learning discriminative spatiotemporal features for precise crop classification from multi-temporal satellite images**. International Journal of Remote Sensing, 41(8):3162–3174, 2020.

112 LA ROSA, L. E. C.; OLIVEIRA, D. A. B. ; FEITOSA, R. Q.. **End-to-end cnn-crfs for multi-date crop classification using multitemporal remote sensing image sequences**. 2021.

113 RUDER, S.. **An overview of multi-task learning in deep neural networks**. arXiv preprint arXiv:1706.05098, 2017.

114 CONGALTON, R. G.; GREEN, K.. **Assessing the accuracy of remotely sensed data: principles and practices**. CRC press, 2008.

115 SANCHES, I. D.; FEITOSA, R. Q.; ACHANCCARAY, P.; MONTIBELLER, B.; LUIZ, A. J. B.; SOARES, M. D.; PRUDENTE, V. H. R.; VIEIRA, D. C. ; MAURANO, L. E. P.. **Lem benchmark database for tropical agricultural remote sensing application**. ISPRS - International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences, XLII-1:387–392, 2018.

116 LOSHCHILOV, I.; HUTTER, F.. **Sgdr: Stochastic gradient descent with warm restarts**. arXiv preprint arXiv:1608.03983, 2016.