Pontifícia Universidade Católica
do Rio de Janeiro

Daliana Lobo Torres

Applying Fully Convolutional Architectures for the Semantic Segmentation of UAV, Airborn, and Satellite Remote Sensing Imagery

Dissertação de Mestrado

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

Advisor : Prof. Raul Queiroz Feitosa
Co-advisor: Prof. José Marcato Junior

Rio de Janeiro
August 2020

Pontifícia Universidade Católica
do Rio de Janeiro

Daliana Lobo Torres

Applying Fully Convolutional Architectures for the Semantic Segmentation of UAV, Airborn, and Satellite Remote Sensing Imagery

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

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

Prof. José Marcato Junior
Co-advisor
Universidade Federal do Mato Grosso do Sul – UFMS

Dr. Guilherme L. Abelha Mota
Universidade do Estado do Rio de Janeiro – UERJ

Dr. Edson Takashi Matsubara
Universidade Federal do Mato Grosso do Sul – UFMS

Rio de Janeiro, August the 14th, 2020

Daliana Lobo Torres

The author received her degree in Biomedical Engineering at the Universidad De Oriente (UO) in 2014. Currently, she has enrolled in the Electrical Engineering master program at PUC-RIO focus on Signal Processing and Control specialization. Since then, she has worked in the field of Digital Image Processing, Remote Sensing, and Machine Learning.

Bibliographic data

Lobo Torres, Daliana

Applying Fully Convolutional Architectures for the Semantic Segmentation of UAV, Airborn, and Satellite Remote Sensing Imagery / Daliana Lobo Torres; advisor: Raul Queiroz Feitosa; co-advisor: José Marcato Junior. – 2020.

v., 78 f: il. color. ; 30 cm

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

Inclui bibliografia

1. Engenharia Elétrica – Teses. 2. Aprendizado Profundo;. 3. Redes Totalmente Convolucionais;. 4. Sensoriamento Remoto;. 5. Segmentação Semântica;. 6. Plataformas de sensoriamento remoto. I. Feitosa, Raul Queiroz. II. Marcato Junior, José . III. Pontifícia Universidade Católica do Rio de Janeiro. Departamento de Engenharia Elétrica. IV. Título.

CDD: 621.3

Acknowledgments

I am truly grateful to my advisor, Prof. Raul Queiroz Feitosa, for the encouragement, his advice, stimulating talks, and generous support throughout my master's studies. I would also like to thank Dr. Patrick Nigri Happ and Prof. Jose Marcato Junior for their support and guidance for the development of this thesis.

I thank my family, my mother, brother, and grandparents, for the dedication and great love, their advice, and supportive words. I wish to thank one of the most lovely, comprehensive, and intelligent men I know: my husband. The best colleague and life partner, thanks for the immense help during this journey.

I would like to thank Pedro Soto and his wife, Brenda Martins for their support and help not only in the academic aspect but also in the personal. Thanks for being there for everything.

I want to thank all my colleagues from the Computer Vision Lab for their companionship and valuable scientific discussions.

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

Abstract

Torres, D. L.; Feitosa, R. Q. (Advisor); Marcato, J. J. (Co-Advisor). **Applying Fully Convolutional Architectures for the Semantic Segmentation of UAV, Airborn, and Satellite Remote Sensing Imagery**. Rio de Janeiro, 2020. 78p. Dissertação de mestrado – Departamento de Engenharia Elétrica, Pontifícia Universidade Católica do Rio de Janeiro.

The increasing availability of remote sensing data has created new opportunities and challenges for monitoring natural and anthropogenic processes on a global scale. In recent years, deep learning techniques have become state of the art in remote sensing data analysis, mainly due to their ability to learn discriminative attributes from large volumes of data automatically. One of the critical problems in image analysis is the semantic segmentation, also known as pixel labeling. It involves assigning a class to each image site. The so-called fully convolutional networks are specifically designed for this task. Recent years have witnessed numerous proposals for fully convolutional network architectures that have been adapted for the segmentation of Earth observation data. The present work evaluates five fully convolutional network architectures that represent the state of the art in semantic segmentation of remote sensing images. The assessment considers data from different platforms: unmanned aerial vehicles, airplanes, and satellites. Three applications are addressed: segmentation of tree species, segmentation of roofs, and deforestation. The performance of the networks is evaluated experimentally in terms of accuracy and the associated computational load. The study also assesses the benefits of using Conditional Random Fields (CRF) as a post-processing step to improve the accuracy of segmentation maps.

Keywords

Deep Learning; Fully Convolution Neural Networks; Remote Sensing; Semantic Segmentation; Remote Sensing Platforms;

Resumo

Torres, D. L.; Feitosa, R. Q.; Marcato, J. J.. **Aplicação de Redes Totalmente Convolucionais para a Segmentação Semântica de Imagens de Drones, Aéreas e Orbitais.**. Rio de Janeiro, 2020. 78p. Dissertação de Mestrado – Departamento de Engenharia Elétrica, Pontifícia Universidade Católica do Rio de Janeiro.

A crescente disponibilidade de dados de sensoriamento remoto vem criando novas oportunidades e desafios em aplicações de monitoramento de processos naturais e antropogénicos em escala global. Nos últimos anos, as técnicas de aprendizado profundo tornaram-se o estado da arte na análise de dados de sensoriamento remoto devido sobretudo à sua capacidade de aprender automaticamente atributos discriminativos a partir de grandes volumes de dados. Um dos problemas chave em análise de imagens é a segmentação semântica, também conhecida como rotulação de pixels. Trata-se de atribuir uma classe a cada sítio de imagem. As chamadas redes totalmente convolucionais de prestam a esta função. Os anos recentes têm testemunhado inúmeras propostas de arquiteturas de redes totalmente convolucionais que têm sido adaptadas para a segmentação de dados de observação da Terra. O presente trabalho avalias cinco arquiteturas de redes totalmente convolucionais que representam o estado da arte em segmentação semântica de imagens de sensoriamento remoto. A avaliação considera dados provenientes de diferentes plataformas: veículos aéreos não tripulados, aeronaves e satélites. Cada um destes dados refere-se a aplicações diferentes: segmentação de espécie arbórea, segmentação de telhados e desmatamento. O desempenho das redes é avaliado experimentalmente em termos de acurácia e da carga computacional associada. O estudo também avalia os benefícios da utilização do Campos Aleatórios Condicionais (CRF) como etapa de pós-processamento para melhorar a acurácia dos mapas de segmentação.

Palavras-chave

Aprendizado Profundo; Redes Totalmente Convolucionais; Sensoriamento Remoto; Segmentação Semântica; Plataformas de sensoriamento remoto

Table of contents

List of figures

List of tables

1
INTRODUCTION

Earth observation via remote sensing technology is widely used to monitor, assess, and gather information concerning the planet's systems. This technology measures emitted and reflected radiation captured by multiple sensors mounted on suitable stable platforms. Each of these platforms is characterized by different technological, operational, and economic factors that play an important role in the sense of how efficiently the information is captured. In this respect, these technologies deliver suitable data, which, combined with the analysis and the development of proper methods, provide important information regarding the natural and anthropogenic environments.

General description of different platforms includes spatial resolution, temporal resolution (revisit time), and object range (distance from the object to the sensor). For instance, satellite surveys can map repeatedly large-coverage areas. However, in general, they have a coarse spatial resolution and may suffer from cloud cover during the image acquisition [3, 4]. Aircraft surveys, on the other hand, offer high-resolution images and tend to have more flexibility since the acquisition time can be controlled under suitable weather conditions. However, they are less stable than space-based technologies and involve expensive cost regarding campaign organization efforts [5]. Unmanned Aerial Vehicles (UAVs), also called Drones, address this cost problem, and offer very high-resolution images which provide a granular understanding of the object(s) in the image [6]. However, the low altitude and short flight endurance currently limit the range of activity of these devices in some applications.

In remote sensing, the mapping of land cover dynamics is a common and challenging problem aiming to locate instances of a given object class in a specific image [7]. On the other hand, computer vision has evolved substantially in the last decade, mainly due to the introduction of deep learning methods. In this context, convolutional neural networks (CNNs) have become the most common approach for different image analysis tasks such as automatic classification, object detection, and semantic segmentation [8–15]. Recently, CNNs have been widely applied for remote sensing problems achieving the state-of-the-art in many applications [16]. Combining the power of CNNs on images from different remote sensing platforms, different studies using object

detection methods were introduced [11, 17, 18]. In this case, a CNN is trained to delineate a bounding box around the target object, providing information about their position and location. Although this information is valuable, some key important details regarding the morphology of the object, like its individual shape or contour, are not provided. In this sense, semantic segmentation based algorithms arise as an alternative to achieve fine-grained information towards complete scene understanding, presenting the potential to capture object forms more accurately than single object detection.

The first idea for deep semantic segmentation methods was to build a patch-based CNN. This approach consists of splitting the image into patches and classifying their central pixel using a traditional CNN. A critical drawback of this method is the redundant operations, specifically in overlapping patches, associated with its high computational cost. To overcome these difficulties, fully convolutional neural networks (FCNs) were first proposed in [19]. The network uses convolutional and pooling layers to build an end-to-end network able to manage different spatial resolutions and predict class labels for all pixels, exploiting context, and location information of the objects in the scene. Later on, with U-Net [20], a technique to improve the spatial accuracy of the segmentation outcome was proposed. Typically, in this approach, the input image is first processed by an encoder stage consisting of convolutional and pooling layers that reduce the spatial resolution. It is then followed by a decoder stage that recovers the original spatial image resolution by using upsampling layers followed by convolutional layers ("up-convolution"). In addition, the network uses the so-called skip connections appending the output of the corresponding layers in the encoder stage to the inputs of the decoder stage. The SegNet architecture [14], as the U-Net, employs the same principle of the encoder and decoder paths. However, instead of using skip connections, the decoder makes use of the pooling indices computed in the pooling operation of the corresponding encoder layers to upsample the result up to the original image resolution.

Some authors proposed the use of a conditional random fields (CRF) based post-processing to further improve the spatial and semantic accuracy of the FCN outcome (e.g., [15, 21]).

Notwithstanding the reported improvements brought about by CRF, these methods have a significant drawback: FCN and CRF need to be trained separately so that such methods constitute no end-to-end solution. In the last few years, real end-to-end FCN architectures for semantic segmentation were published, which reportedly performed at least as good as prior solutions that included CRF post-processing (e.g., [15, 22]). This was achieved due to

innovative techniques to capture multi-scale context within the FCN, such as global-to-local contexts aggregation as in ScasNet [22] and atrous spatial pyramid pooling in DeepLabv3+ [23].

Motivated by the state-of-the-art of DL semantic segmentation models, we evaluate in this work the use of five state-of-the-art deep learning methods for the semantic segmentation of individual targets on three different scenarios: segmentation of a single tree species, of building rooftop and deforestation in the Amazon Biome using images derived from UAV, aerial and satellite images.

Regarding satellite imagery, we focus on the monitoring of environmental changes, specifically on the deforestation practices within the Brazilian Amazon. The implications of the Amazon deforestation regarding global warming and the conservation of ecosystems and biodiversity of the region constitute a global concern for different government associations. In this sense, some organizations such as the National Institute for Space Research (INPE) focus its effort on the processes of understanding and monitor the rhythms of environmental changes in the Amazon Biome. Yet, it is still lacking of automatic methods for the effective mapping and estimation of annual deforestation rates.

On the other hand, the usage of aerial imagery for mapping the location and morphology of building rooftop in dense urban environments can be a valuable input for policymakers. Reliable and timely maps of buildings are essential for urban planning and city modeling, especially in areas experiencing rapid urbanization, evidencing the possibility of the application of different FCN approaches in this research area.

Concerning UAV platform, we focus on identifying the canopy of the threatened species *Dipteryx alata* Vogel, also known as cumbaru. It comes about in midwestern Brazil, and due to its particular shadow and canopy, it is used for afforestation practices over urban areas. This species has a tremendous social and economic relevance for the development of some areas of the Brazilian Cerrado [24]. It has been threatened by extinction according to the IUCN (2020) (The International Union for Conservation of Nature's Red List of Threatened Species, `https://www.iucnredlist.org/species/32984/9741012`), which makes its preservation a very important issue since this particular species provides fruits for a large number of bird species.

1.1 Objectives

The objectives of this work are the following:

– **General objective:**
Evaluate deep learning methods to segment individual objects using images from different platforms (Satellite, Aircraft and UAV systems).

– **Specific objectives:**

1. Compare five state-of-the-art deep learning semantic segmentation methods, namely U-Net, SegNet, Fully Convolutional DenseNet, and Deeplabv3+ with the Xception and MobileNetV2 backbone.
2. Assess the performance on the aforementioned networks for the segmentation of three object classes: individual tree species, rooftops and deforested areas.
3. Assess the performance of the aforementioned networks for the semantic segmentation of UAV, aerial and satellite images.
4. Assess the improvements of using CRFs as a post-processing step for the Cumbaru and Building Rooftop datasets.

1.2 Contributions and Novelties

The main contributions of this work are the following:

1. A comparison of five state-of-the-art fully convolutional neural networks, namely U-Net, SegNet, Fully Convolutional DenseNet, and Deeplabv3+ with the Xception and MobileNetV2 backbone.

2. An analysis of the aforementioned FCNs on differents platforms: UAV, airborne, and satellite images.

3. An analysis of the FCNs in the semantic segmentation of three object targets: individual tree species, rooftops, and deforested areas.

4. An assessment of Conditional Random Fields as post-processing in the Cumbaru and Rooftop datasets.

1.3 Organization of the remainder text

Chapter 2 describes the works related to this research.

Chapter 3 provides the theory and fundamental concepts essential for the understanding of the tested methods.

Chapter 4 describes the study areas and introduces the fundamentals of FCNs, specifically, the FCN approaches investigated in this work.

Chapter 5 further presents the protocol followed in our experimental analysis and the recorded results.

Chapter 6 summarizes the conclusions derived from the performed experiments and provides directions for the continuation of this research.

2
RELATED WORKS

This chapter summarizes the works related to this research that have been proposed so far. Most of them were developed in the context of deforestation detection, rooftop segmentation, and tree species segmentation.

2.1
Deforestation Detection

Change detection is defined as the process to analyze and measure the differences in a region at different dates. In particular, for the forest context, change detection has been established as an essential task for monitoring the degradation of forest environments and for the preservation of natural ecosystems. Several studies are available in the literature to detect and monitor forest changes relying on satellite images, mainly due to its optimal spectral, spatial, and temporal properties [25–28]. In this regard, multiple change detection techniques have been developed [29–32], which constitute an effective instrument in the processes of identifying and labeling regions that underwent significant changes.

Visual interpretation constitutes one of the earliest techniques for monitoring deforestation. In this case, the identification of forest change is carried out by the analysis of different factors such as color, tonality, texture, shape, and context [32]. Using the combination of these elements, Asner et al. [32] incorporates reflectance data and texture analysis to assess forest canopy damage in the Amazon using Landsat images. The main disadvantage of this technique is the considerable time needed to identify the changes, being in some cases not feasible in the analysis of large areas.

In comparison with visual interpretation, automatic approaches allow a faster analysis. Some of the early approaches includes techniques based on image algebra [29, 31, 33, 34]. Nelson et al. [35] studied image differencing, image rationing, and vegetation index differencing (VZD) for detecting forest canopy alteration and found that the VZD shows more accurate results delineating healthy and defoliated forest. Hayes et al. [36] assessed the normalized difference vegetation index (NDVI) image differencing, principal component analysis (PCA), and RGB-NDVI change detection to detect deforestation practices in

Guatemala's Maya Biosphere Reserve. The authors found that the RGB–NDVI reached the highest accuracy of about 85%.

Several authors reached different conclusions about which method yielded the best results among the different algebra-based approaches since these results differ depending on the characteristics of the research areas. Moreover, the classification step to differentiate change from no-change depends on the selection of an optimal threshold, being necessary some pre-classification techniques to define the best value.

Other methods of change detection include machine learning algorithms to perform direct classification. Some of these works are based on more complex systems such as artificial neural networks [37], decision trees [38], fuzzy theory[39], and support vector machines [40]. These change detection methods present good results for identifying changes in medium- to coarse-resolution imagery, but fail when dealing with high-resolution images because they tend to produce the salt-and-pepper pattern on the resulting maps [38]. This effect occurs due to high-frequency components and high contrast of the high-resolution images, as well as the variation in the image acquisition, which often results in too many changes being detected [38]. Indeed, the main limitation of the method is the difficulty of modeling contextual information as the pixels within a neighborhood are often ignored [41].

Deep Learning methods have shown great potential for change detection approaches, showing higher results in comparison to traditional machine learning methods [42]. The capability to capture spectral-spatial-temporal information in an image sequence makes deep learning approaches very suitable to the change detection task [43–46]. In the category of pixel classification, Hou et al. [47] use low-rank saliency computation based on the premise that only small regions of the image changed. Here, a CNN is applied to the change features extracted from super-pixels generated using the SLIC algorithm. Other approaches include patch-based classification, which includes the Early Fusion and Siamese CNN [48–50].

Fully convolutional neural networks (FCNs) are one of the leading types of architectures within the different deep learning networks. Unlike traditional CNNs, which predict a unique probability for each input image, FCNs can assign probabilities to each pixel in the image. One of the first works using FCN in the change detection context was proposed in [51]. In this work, earlier proposed ideas of Early Fusion and Siamese Network were extended, by training from scratch FCNs instead of the traditional CNNs, achieving better performance and faster computation. Recently, other fully convolutional methods such as the U-Net have been successfully applied for the automatic

mapping of bi-temporal images [52, 53]. De Bem et al. [53] compare different architectures such as SharpMask, U-Net, and ResUnet to map deforestation between images on different dates. The ResUnet model achieved the best results in terms of F1-score and IoU, while SharpMask and U-Net models presented comparable but slightly lower results. Peng et al. [54] proposed an effective architecture named UNet++ based on the U-Net model, dense skip connections, and residual blocks. The author concludes that the proposed UNet++ outperforms the results in [53] on both visual and quantitative metrics. Motivated by this, we then propose the use of other state-of-the-art FCNs architectures following the Early Fusion approach to detect deforestation changes in the Brazilian Amazon.

2.2 Building Rooftop Segmentation

The automatic mapping of different urban objects such as buildings plays an important role in environmental modeling and monitoring, for updating geographical databases, disaster managing, land cover, infrastructure planning, and counting. In this regard, several algorithms ranging from traditional methods to more advanced machine learning approaches have been suggested and widely used for remote sensing images.

During the early days, these techniques were mainly used for the image classification task due to the lack of high-resolution images. In this sense, Bischof et al. [55] compared the results of the maximum likelihood classifier with an artificial neural network (ANN) for classifying satellite images into forest, buildings, agricultural land, and water. Later, some researches in the area focused on combining the potentialities of ANN with other methods to improve the image classification results also of satellite images. In this line, Abraham et al. [56] proposed an automatic technique based on a wavelet-based watershed method and ANN for building classification on highly dense urban areas. Further on, Joshi et al. [57] proposed a method for the automatic detection of building rooftops in satellite images. Here, a traditional ANN is first trained to perform a binary classification between rooftop and non-rooftop regions, following a Support Vector Machine (SVM) classifier to improve the performance of the model and reduce false-positives that could be left by the previous step. One weakness of this approach was the few cases where the ANN was very accurate. Moreover, the method fails to detect rooftops with different color information.

With the advances in the computer vision field, feature extraction through CNNs has outperformed traditional approaches for visual recognition

concerning roof shapes. In this regard, Castagno et al. [18] used different CNN based architectures such as Resnet50, Inceptionv3, and Inception-ResNet to extract high-level features of Satellite and airborne LiDAR imagery. Similar to[57], these features then fed a second-stage SVM or random forest classifiers to provide an accurate single roof geometry decision. Satellite image and LiDAR data fusion achieved the highest accuracy .The good performance was achieved largely due to the fusion of optical images with data LiDAR.

Many studies show the capabilities of CNNs as a feature extractor for object-based classification [18, 57–59]. These studies, in fact, have been shown to achieve good overall performance. Although highly dependent on the intermediate classification outcomes of CNNs, they do not constitute end-to-end solutions.

In the last few years, advances in remote sensing platforms have made available large amounts of high-resolution aerial imagery making room for different applications of deep learning methods [60, 61]. In particular, for the semantic segmentation task, different methods have shown good performance in the segmentation of different objects. In this regard, Zhong et al. [62] proposed the use of different configurations of FCN to segment buildings and roads using aerial RGB images, reaching a precision accuracy of about 78%. Further on, Xu et al. [63] improved the building classification by the application of a new model based on residual networks, defined as Res-U-Net, to perform segmentation of high-resolution imagery. On the other hand, Wu et al. [64] proposed a multi-constraint fully convolutional network (MC–FCN) for the automatic segmentation of buildings on high-resolution aerial images, achieving improvement over the U-Net of about 3.2% in the Jaccard index at the cost of 1.8% increment in the training time. Recently, Lichao et al. [65] proposed a relational context-aware fully convolutional network on two aerial images datasets: ISPRS Vaihingen and Potsdam benchmarks. The proposed method achieved competitive results and outperformed the baselines in terms of mean F1-score, mean IoU, and overall accuracy.

Some researches try to improve the semantic segmentation outcome by using a post-processing step based on Conditional Random Field (CRF). Mnih et al. [66] investigated the use of deep neural networks trained on aerial images to learn discriminative features in the detection of roads and buildings from noisy labels. Moreover, it also proposed the use of the CRF technique for improving the predictions of the networks, achieving moderate results. Kluckner et al. [67] proposed a segmentation technique based on super-pixels and a CRF stage to classify and to construct synthetic 3D models of rooftops on aerial images. The results showed that the integration of the

CRF stage to the super-pixel method improved the final building classification significantly. In the context of the ISPRS 2D semantic labeling benchmark, Gerke et al. [68] performed a multi-class image classification by training an Adaboost-based classifier followed by a CRF. The CRF stage smooths the Adaboost prediction, which can affect negatively the final classification result. A later ISPRS benchmark [69] used CNNs and hand-crafted features to perform a pixel-wise classification of aerial imagery, and a CRF to improve the segmentation accuracy. The reported results showed no significant accuracy gains due to the CRF step, although the improvement of the visual results may make it worthwhile. One downside of the method is that CRF tends to reduce regions with ambiguous probabilities and eliminate small labeled regions [69].

In this work, we compare and analyze five deep fully convolutional networks, in the semantic segmentation of individual building rooftops on aerial images. Secondly, CRFs post-processing is performed on the resultant segmentation maps in the final classification.

2.3 Single Tree Species Segmentation

Forest monitoring provides essential information to support public policies related to protection, control, climate change mitigation, and sustainable development. Therefore, the continuous monitoring of forest trends through remote sensing enables a cost efficient measurement of vegetated ecosystems. In this context, satellite observations constitute a suitable platform to cover large areas at regular periodicity [70].

In the forest monitoring context, single tree detection is an essential task for many applications, including resource inventories, wildlife habitat mapping, biodiversity assessment, and hazard and stress management [8]. Over the years, researchers have worked in this field, mapping single tree species based on different satellite imagery and achieving moderate results [9, 10, 71–73]. In the last decade, new approaches emerged to take advantage of the characteristics of active sensors, especially light detection and ranging (LiDAR) systems, which became a trend for tree crown detection [74]. More recently, the authors in [75] concluded that combining LiDAR data with optical imagery generally leads to better classification accuracy. Although this conclusion might be generalized, the authors focused on classifying tree species in urban environments.

In fact, urban forests are a particular case of forests with singular attributes and peculiarities. Urban forests are commonly defined as woody vegetation located in an urban area and usually limited to single and/or groups of trees distributed in parking places, gardens, small parks, and along roads

in the city. Thus, the heterogeneity of urban environments makes the accurate classification of tree species more challenging than in natural forests. Firstly, a high spatial resolution image is required in order to differentiate them as individual objects. Secondly, with the progress of urbanization, urban trees are heavily influenced in their environment by urban patterns like streets, communities, and factories [76].

Unmanned Aerial Vehicles (UAVs) can provide appropriate temporal and spatial resolution images to produce suitable datasets for mapping forested areas on the individual tree level [77]. This may allow a better detection of single trees in urban scenarios. Following this trend, Feng and Li [78] proposed a method for mapping tree species in urban areas based on histograms and thresholding using UAV observations. Similarly, Baena et al. [79] used object based image analysis on high spatial resolution UAV images to identify and quantify tree species across different landscapes.

On the other hand, some deep learning based approaches for tree species detection have been proposed in recent years. Li et al. [80] presented a deep learning based framework for oil palm tree detection and counting, using high spatial resolution satellite images. Weinstein et al. [81] used RGB images from an airborne observation platform along with airborne LiDAR data to detect tree crowns through a deep learning network.

Considering UAV platforms, Natesan et al. [82] proposed a deep learning framework for tree species classification. In this approach, images of pre-delineated tree crowns were the inputs to a CNN to classify the delineated trees to one out of three classes: red pine, white pine, and non-pine. Similarly, Masanori et al. [83] used UAVs to acquire RGB images of individual tree crowns and carried out a multiresolution segmentation algorithm [84] to classify seven different types of trees. Overall accuracy up to 89% was reported in this study. In [11], Santos et al. proposed different deep learning methods for detecting law protected tree species using high resolution RGB imagery. These methods delivered a bounding box that enclosed each object instance, but did not delineate the shape or contour of the target.

In recent years, a few studies have already evaluated the potential of the FCN architectures, specifically U-Net, for forest mapping from optical images [85, 86]. In [85], the authors used a U-Net to identify instances of a given tree species from WorldView-3 images. Similarly, in [86], the U-Net was trained with the RGB bands and the digital elevation models (DEM) from high resolution UAV imagery. The importance of monitoring urban forests and the lack of studies on using FCNs' capabilities for this purpose motivated the present study.

3
FUNDAMENTALS

Remote Sensing Image analysis plays a key role in the earth observation technology. In recent years, as deep learning methods emerge, remote sensing image classification has gained recognition offering novel possibilities for the research in different applications. In this chapter, we first present an overview of the fundamentals of remote sensing techniques and their different platforms. Moreover, we also introduce the fundamental principles of different deep learning methods, specifically Fully Convolutional Networks.

3.1
Remote Sensing

Remote sensing is the science of detecting and monitoring the physical characteristics of an area by measuring the emitted or reflected electromagnetic radiation from the Earth's at distance [87]. These remote sensors provide fast and repetitive image acquisition of extensively large areas displacing the slower, costly data collection on the ground. Remote sensing has become a well-established tool for many applications, ranging from monitoring forest fires [28], deforestation [88], climate changes [28] and others, providing a global perspective of the earth.

The electromagnetic energy can be recorded by either passive or active sensors (see Fig 3.1). Depending on what is being sensed and the application, these various sensors are mounted on different platforms for the remote image acquisition.

Passive sensors are devices that measure the naturally available energy reflected by the Earth surface. A common example of passive sensor-based technologies is the optical sensors that collect reflected electromagnetic energy generated by the sunlight. Nonetheless, important drawbacks of these sensors include their dependency on the availability of natural energy, the presence of clouds, and poor weather conditions. Other types of natural energy sensed by passive sensors include thermal infrared and radiometers [89].

On the other hand, active sensors incorporate the source of electromagnetic radiation and do not depend on the sun. Specifically, they emit radiations toward the target and measure the time delay between the emission and the

radiation reflection from the target [90]. Some examples of active sensors are RADAR (Radio Detection And Ranging), LiDAR (Light Detection And Ranging), and SAR (Synthetic Aperture Radar).

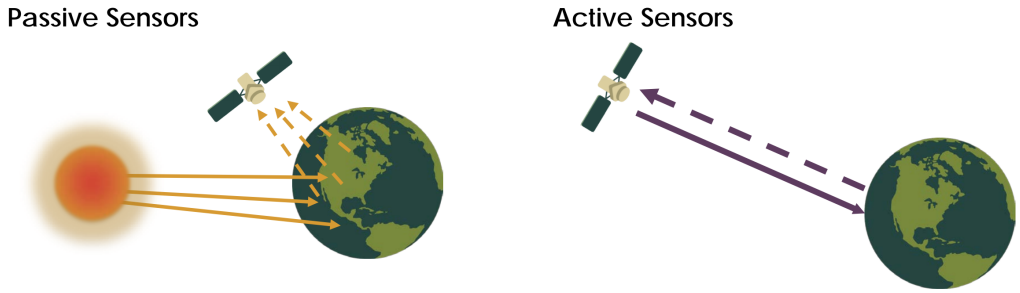


Figure 3.1: Passive and Active sensors

3.1.1 Remote Sensing Platforms

Different platforms are used as vehicles from which remote sensing sensors map and monitor the Earth. Typical platforms are aircrafts (plane, unmanned aerial vehicles (UAVs), helicopters, balloons, etc) or on a spacecraft (satellites) orbiting the Earth from outside the atmosphere. Each platform has advantages and disadvantages in terms of distance from sensor to the object of interest, periodicity of image acquisition, timing of image acquisition, location and extent of coverage, as illustrated in Figure 3.2.

Figure 3.2: Remote Sensing Platforms. From [91]

Airborne Platforms

Airborne platforms include airplanes, helicopters, unmanned aerial systems (UAS), and free-floating balloons. These vehicles can be categorized

based on their elevation restrictions in low (less than 30,000 feet) and high altitudes flights (above 30,000 feet). In general, the higher the fly, the more stable a platform is at the cost of less resolution and detail of the image. The main advantage of airborne remote sensing, compared to satellite, is the ability to offer high spatial resolution imagery (20 cm or less).

UAV is a kind of aircraft system that is operated remotely by automatic systems or from the ground. UAVs were originated mostly for military applications. Now they can be used to collect remotely sensed data for a variety of commercial and research purposes. The primary advantages of UAV sensors are the very high-resolution images and its low altitude flights to capture detailed information of the terrain. It can also allow a more precise analysis of the landscape, regarding the local-scale analysis about the area of interest [28]. This characteristic enhance the suitability of UAVs for applications in Precision Agriculture [92]. The primary limitation of this technique is the turbulence and variable wind speeds causing in some cases blurred images.

Satellite

Airborne remote sensing missions are often carried out as one-time operations, whereas earth observation satellites offer the possibility of the continuous monitoring of the Earth. The path of the satellite is referred to as its orbit. In this regard, there exist satellites following north-south orbits or polar orbits or in conjunction with the Earth's rotation (west-east orbit). Many of these orbits are also sun-synchronous so that the satellite covers the same area at the same local time when solar illumination is available. One example is the Landsat 8 satellite, which orbits the Earth in a sun-synchronous way with a temporal resolution of 16 days. On the other hand, as satellites are positioned at a high altitude (705 km for Landsat 8) the spatial resolution is not as good as the airborne platforms. Landsat data, for example, has 30m spatial resolution, meaning that each pixel corresponds to a 30m $\times$ 30m large area on the ground.

Satellite platforms can image any place on Earth with the same quality and standard of service. Many remote sensing applications have benefited from currently free available satellite images and from historical imagery.

3.2 Convolutional Neural Network

One of the most powerful Artificial Neural Networks architecture is the Convolutional Neural Networks (CNNs). CNNs circumvent the limitations of traditional neural networks in terms of computational complexity when operating on large images. Comparatively, CNNs involve less parameters, an operate on local regions instead of the image as a whole [93].

3.2.1 Topology of Convolutional Neural Network

The **input** of traditional CNNs is a 3D input tensor. Its *width* and *height* define the spatial dimension of the image, and the *depth* the numbers of channels. A typical architecture of a convolutional network is made of three basic layers: convolutional layers, pooling layer, and fully-connected layer, as shown in Figure 3.3.

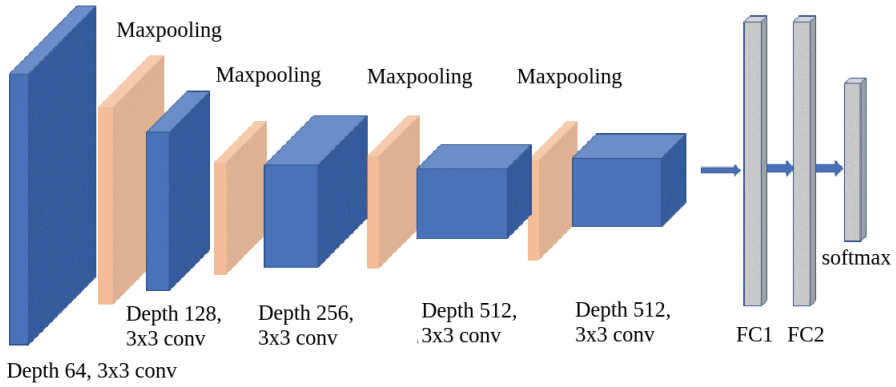


Figure 3.3: VGG-16 architecture. Adapted from [1, 2]

Convolutional layers extract feature representations of each input feature map. Conventionally, the first convolutional layers capture low-level features such as edges, color, while deeper in the network, layers capture more abstract representations or high-level features.

Each convolution layer is equipped with several convolution kernels. The scalar product between the learnable weights of the kernels and a region of the input image defined by its field of view is first calculated. Then, the result is applied to a nonlinear activation function, such as sigmoid, hyperbolic tangent, softmax, or RELU, to obtain a new feature map. The spatial dimension of the output feature map will depend on the kernel size, on the adopted stride, and on the use or not of padding.

Pooling layers reduce the spatial dimension of the input image as well as the number of parameters and computational load involved in the subsequent layers. Pooling layers come about typically in two variants: the Max-Pooling and the Average-Pooling. The Max-Pooling operation returns the maximum value of a sub-region of the feature map. Average-Pooling, on the other hand, computes the average over each sub-region. Average and Max-Pooling are different in the amount of information they retain, while Max-pooling eliminates less important elements, average pooling blends them. This could be preferable in situations where no dominant feature information is required.

After feature extraction, the final classification of the image is done using **fully connected layers**. The feature map produced by the previous layer is flattened into a vector, which is first multiplied by a weight matrix and then applied to a non-linear activation function, which delivers the final probabilities for each class of the problem.

Normalizing the input data of neural networks has been known for decades [29] to be beneficial to neural network training. There are many reasons for that. One of them is to prevent the early saturation of the non-linear functions, gradient oscillations, and convergence problems. However, even with normalization, these problems can occur during the training of deep neural networks because the distribution of the intermediate results changes as the weights are updated during the training. Ioffe et al. [94] refers to this issue as Internal Covariate Shift, and address the problem by normalizing the results of intermediate layers within the network for each training mini-batch. This operation is called **batch-normalization**.

During training, batch-normalization normalizes the output of a previous activation layer by subtracting the batch mean and dividing by the batch standard deviation. The standardized output can be shifted and scaled using two trainable parameters β and γ which define the new mean and standard deviation of each input variable per mini-batch. Such as transform is differentiable which allows the backpropagation of the gradient through the network. In the inference time, γ and β are fixed, using the previous calculated means and variance of each batch.

3.3 Fully Convolutional Neural Network

Over the past few years, the fully convolutional networks (FCN) [19] have gained recognition due to their ability to perform pixel-based classification in an end-to-end fashion [95, 96].

This task is known as semantic image segmentation, and can be efficiently acomplished by modifying the fully-connected layer of a traditional CNN into convolutional layers [19]. Typically, these networks consist of an encoder module, which reduces spatial resolution by convolution and pooling operations through consecutive layers, and a decoder module that retrieves the original spatial resolution. Similar to conventional CNNs, convolutional layers extract meaningful features by convolving the input image with kernels or filters. During convolution, each filter operates over a local region of the input volume, which is equivalent to the filter size. The spatial extent of the input image considered in calculating a position of an activation map is called receptive field or field of view. To enlarge the receptive field, we can use larger filters or add more layers. Both strategies imply more parameters, more operations, and higher computational complexity. To compensate for this effect and reduce the computational cost, pooling layers reduce the resolution of the feature maps. In consequence, part of spatial information gets lost, mainly at fine details.

3.3.1 Atrous Convolutions

To increase the field of view without increasing the number of parameters, Chen et al. [97] proposed the atrous convolutions. Atrous convolution, also known as dilated convolution, operates on an input feature map ($\mathbf{x}$) as follows:

$$\mathbf{y}(i) = \sum_j \mathbf{x}[i + r * j]\mathbf{w}[j] \tag{3-1}$$

where i is the location in the output feature map $\mathbf{y}$, $\mathbf{w}$ is a convolution filter, and r is the dilation rate that determines the stride in which the input signal is sampled [98].

The basic idea consists in expanding a filter by including zeros between the kernel elements. For example, if a $k \times k$ filter is expanded by an expansion rate r, r-1 zeros are inserted between each adjacent element of the original filter along each dimension. Thus, the receptive field is expanded to $[k + (k - 1)(r-1)] \times [k+(k-1)(r-1)]$ [99] (see Figure 3.4). In this way, we increase the receptive field of the output layers without increasing the number of learnable kernel elements and the computational effort.

Later, Chen et al [15] proposed the Atrous Spatial Pyramid Pooling (ASPP) illustrated in Figure 3.4.

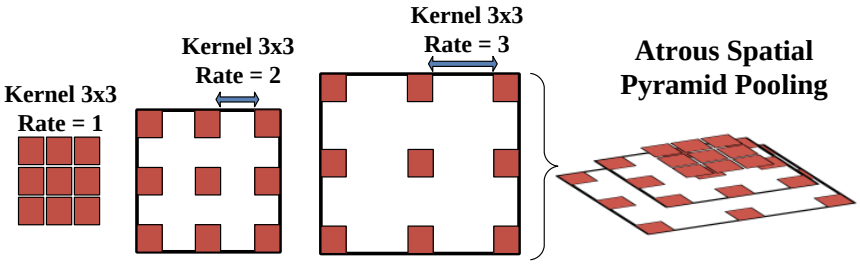


Figure 3.4: Atrous Spatial Pyramid Pooling.

This technique involves employing atrous convolution with different rates in parallel as a strategy to extract features at multiple scales and to alleviate the loss of the spatial information intrinsic of pooling or convolutions with striding operations [23]. Notice that, the receptive field gets larger with increasing rates while maintaining the number of parameters [99].

3.3.2 Depthwise Separable Convolution

Another way to reduce computational complexity and the number of parameters in a model is through separable convolutions. Recent deep learning approaches such as [98, 100] apply separable convolution to both, the encoder and the decoder stages in the FCN model. Conceptually, the spatial separable convolution brakes down the convolution into two separate operations: a depthwise and a pointwise convolution, as illustrated in Figure 3.5.

In the traditional convolution, the kernel is as deep as the input and operates on all input channels. A depthwise separable convolution involves two steps. First, a spatial convolution is carried out independently over each input channel. Here, the number of filters is equal to the number of input channels and each filter operates independently on a single channel [100]. After completing the depthwise convolution, a so-called pointwise convolution performs a 1×1 convolution with a kernel as deep as the number of chanels. The main advantage of depthwise separable convolutions is that they involve fewer parameters compared to regular convolutions, implying fewer operations and faster computation [100].

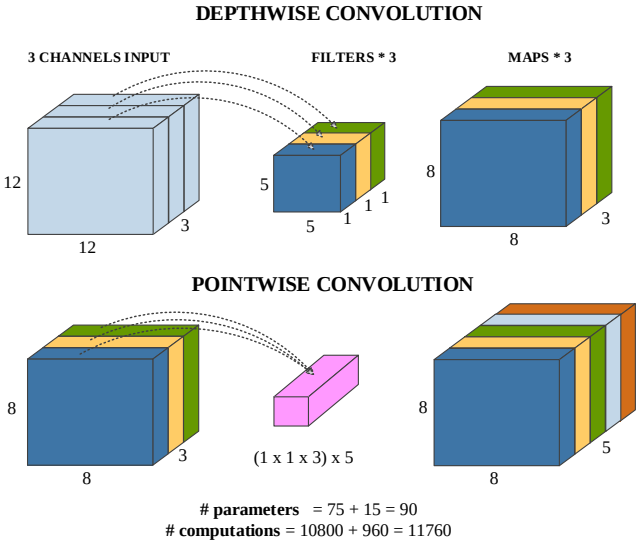


Figure 3.5: Depthwise Separable Convolution.

4
METHODS

In this section, we describe the four FCN architectures assessed in this work: SegNet, U-Net, FC-DenseNet, and the two variants of the DeepLabv3+ related to the adopted backbone: Xception and MobileNetV2. Finally, we revisit the idea of applying conditional random fields (CRFs) as a post-processing technique to improve the overall segmentation outcome.

4.1
U-Net

Like any traditional fully convolutional network, the U-Net (see Figure 4.1) has an encoder-decoder architecture [20]. The encoder is a stack of convolutional and max-pooling layers. The decoder is a symmetric expanding path that uses learnable deconvolution filters to upsample the feature maps. The main novelty introduced by this network is the so-called skip connections. Specifically, they allow the concatenation of the output of the transposed convolution layers with the correspondent feature maps of the encoder stage [101]. This step aims at retrieving the fine characteristics learned by the contracting stages to restore the original input image's spatial resolution [20].

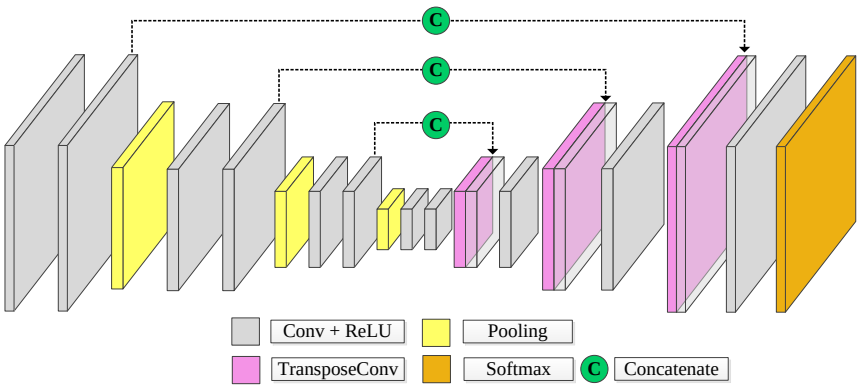


Figure 4.1: Unet

4.2 SegNet

SegNet was also one of the early proposed FCN architectures for semantic segmentation [14]. This network has an encoder and a corresponding decoder path, followed by a final pixel-wise classification layer. The encoder comprises a series of convolutional layers, whose outputs are normalized before being applied to a nonlinear activation function followed by 2×2 max-pooling (see Figure 4.2). A distinguishing characteristic of SegNet is that it keeps the pooling indices, i.e., the position of the cell within each 2×2 group of pixels where the max-pooling operation took the maximum from. These indices are forwarded to the correspondent upsampling layer of the decoder stage. Each upsampling step of the decoder stage involves doubling the spatial resolution. The max-pooling indices stored during the encoder phase determine the cell of the corresponding 2×2 array at the higher resolution output where each input value is to be loaded. The other three cells of the 2×2 array are zeroed. In this way, SegNet seeks to retrieve the input image details lost in the encoder downsampling steps.

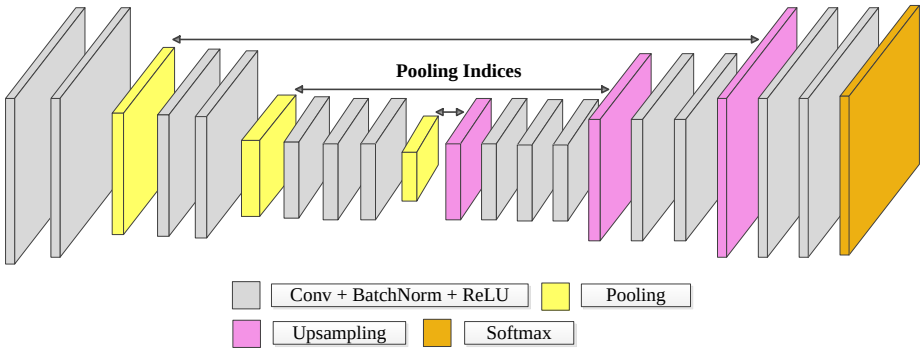


Figure 4.2: SegNet architecture.

4.3 FC-DenseNet

Based on fully convolutional networks, Jégou et al. [102] extended the DenseNet network [103] by adding an upsampling path to recover the input resolution and proposed the fully convolutional DenseNet (FC-DenseNet). Its architecture is illustrated in Figure 4.3. The traditional DenseNet is built on the so-called dense blocks. Each dense block layer is composed of batch-normalization, followed by a ReLU activation function and a 3×3 convolution [102]. The output of a dense block is the concatenation of the outputs of each

layer in the current block. Thus, the number of feature maps increases after each layer, by a factor of k, a network hyperparameter called growth rate.

FC-DenseNet keeps the dense blocks of Smith et al. [103] and includes the downsampling and upsampling paths with skip connections. The downsampling path consists of dense blocks followed by transition down (downsampling) layers, which are composed of a batch normalization, ReLU activation function, an 1×1 convolutional layer, and a 2×2 max-pooling operation [102]. Analogously, the decoder consists of dense blocks and transition up (upsampling) layers, which perform a single transposed convolution with stride 2 [104]. Like the U-Net, the skip connections concatenate the feature maps in the upsampling path with the downsampling feature map at the same level. To avoid an excessive growth of feature maps in the upsampling path, the input of the dense block is not concatenated with its output [102].

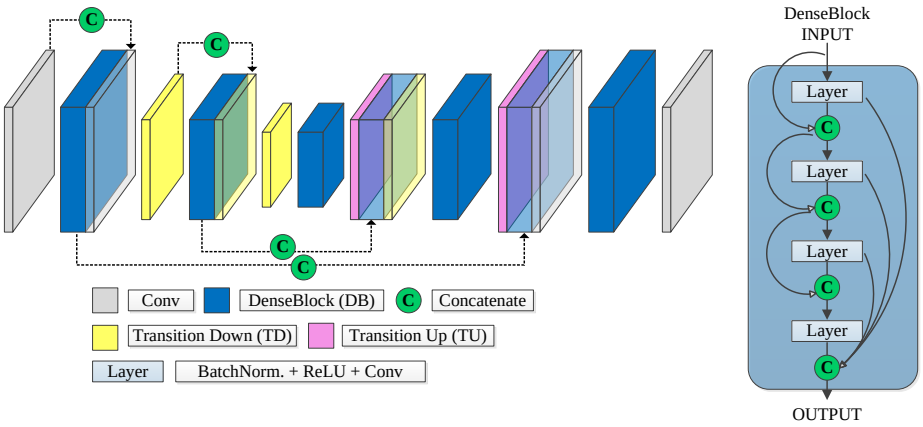


Figure 4.3: FC-DenseNet architecture.

4.4 DeepLabv3+ with the Xception Backbone

Keeping the encoder-decoder structure, the fourth and fifth approaches considered in this work are based on the DeepLabv3+, which represents the state-of-the-art for semantic image segmentation [98] at the time this dissertation was written. A characteristic of this method is the atrous spatial pyramid pooling (ASPP) described in the Section 3.3.1. In relation to conventional architectures, ASPP allows increasing the field of view and thus the spatial context considered at each layer with a smaller increase in the number of parameters and in computational complexity

DeepLabv3+ inherited the depthwise separable convolutions (described in section 3.3.2) introduced in its predecessor, the DeepLabv3 version. Furthermore, the DeepLabv3+ version modified the Xception model presented in [100], including more layers, replacing all max-pooling operations by depthwise separable convolutions, and adding batch normalization and ReLU activation after each 3×3 depthwise convolution[98], as described in Figure 4.4.

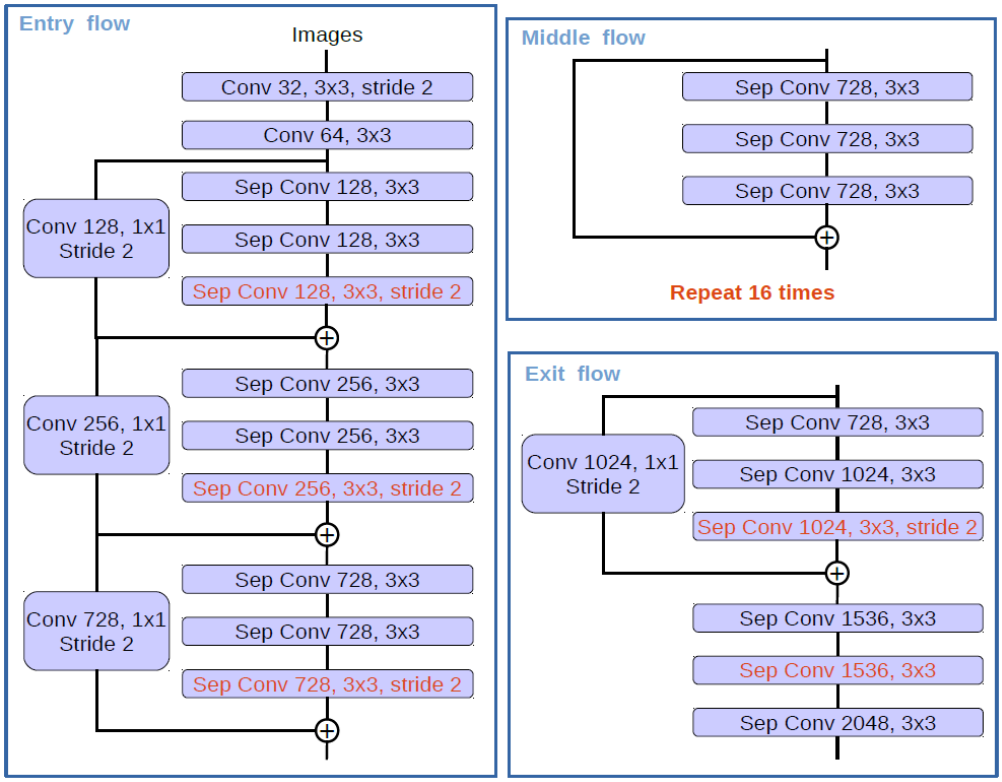


Figure 4.4: DeepLabv3+ Xception backbone Encoder [98]

In the decoder stage, the features obtained from the encoder are upsampled by a factor of 4 and then concatenated with the corresponding low-level features [105]. To make better use of higher level semantic features extracted by the encoder, an 1 × 1 convolution is employed to reduce the number of channels. After the concatenation, a 3×3 convolution is applied to refine the features, ending with another bilinear upsampling by a factor of 4 to obtain the resolution of the input image [98, 106]; see Figure 4.5.

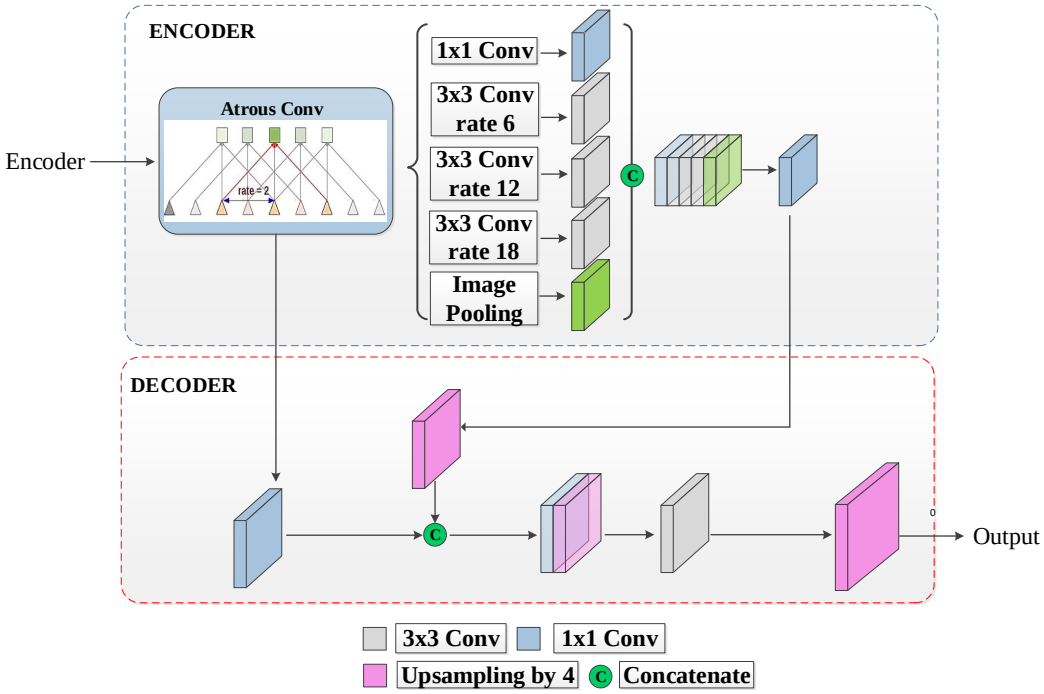


Figure 4.5: DeepLabv3+ architecture.

4.5 DeepLabv3+ with the MobileNetV2 Backbone

We also evaluated a variant of the DeepLabv3+ using a MobileNetV2 backbone [107]. This model was proposed to reduce computational complexity so that DeepLabv3+ could run on mobile devices. The key concept behind MobileNetV2 is the use of inverted residual blocks in the bottleneck of the main architecture. In conventional residual blocks, the depth of the tensor comprising the input feature maps is first reduced by a 1×1 convolution whose output feeds a subsequent 3×3 convolution. Prior to adding the result to the input feature map, another 1×1 convolution is carried out to match the depth of input feature maps.

The inverted residual block presented in [107] works the other way around. It first applies a 1×1 convolution to increase the depth of feature maps' tensor, followed by a 3×3 depthwise convolution. A subsequent 1×1 convolution compresses the resulting tensor back to the depth of the input feature map. This scheme involves considerably fewer parameters than the conventional residual block and is more computationally efficient.

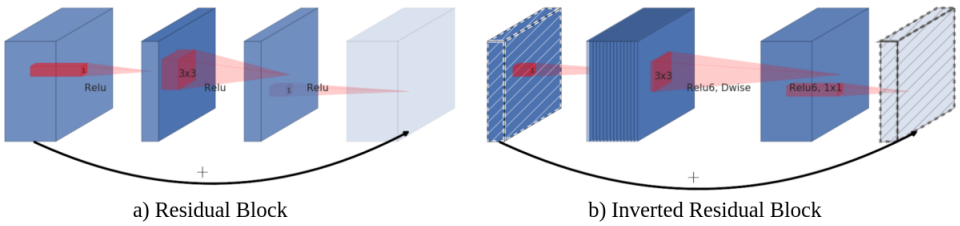


Figure 4.6: Difference between Residual Block and Inverted Residual Block [107].

The encoder architecture of DeepLabv3+ with the MobileNetV2 backbone starts with a convolution with 32 filters, followed by a stack of inverted residual blocks (IRB) [107], as described in the Table 4.1. When the table indicates that the inverted residual block is repeated n times, the stride specified in the table is applied only to the first block. For all other blocks the stride is equal to 1. The expansion factor, on the other hand, stands for the number of times the number of channels in the input tensor of the IRB is going to increase.

Table 4.1: Encoder of DeepLabv3+ Mobilenetv2 architecture

Operation	**No. channels**	**Expansion factor**	**Times**	**Stride**
3×3 Conv	32	-	1	1
IRB	16	1	1	1
IRB	24	6	2	2
IRB	32	6	3	2
IRB	64	6	4	1
IRB	96	6	3	1
IRB	160	6	3	1
IRB	320	6	1	1

4.6 Conditional Random Fields

To improve semantic segmentation and labeling accuracy, probabilistic graphical models have been used as post-processing. Markov random fields (MRFs) and particularly conditional random fields (CRFs) have achieved widespread success in this task [108–110].

While deep neural networks have proven efficient in learning features from a small field of view, they fail to capture global context information. To

address this issue, approaches have been proposed to combine the effectiveness of CNNs to learn discriminatory features, with the CRF's ability to model broad spatial contexts. CRF approaches semantic labeling as a probabilistic inference problem assuming that neighboring pixels tend to share the same class label unless their descriptors differ significantly.

Given the set of pixels $i \in S$ of an image, let $\mathbf{x} = \{\mathbf{x}_i\}_{i \in S}$ be the observed data and $\mathbf{y} = \{y_i\}_{i \in S}$ its corresponding labels, where y_i may take values in $\{l_1, ..., l_m\}$ and m is the number of available classes. A CRF models the posterior probability $P(\mathbf{y}|\mathbf{x})$ of the set of labels $\mathbf{y}$ given the image data $\mathbf{x}$ as follows:

$$P(\mathbf{y}|\mathbf{x}) \propto \exp\left\{ -\left[\sum_{i \in S} A(y_i, \mathbf{x}) + \sum_{i \in S} \sum_{j \in N_i} I(y_i, y_j, \mathbf{x}) \right] \right\}, \tag{4-1}$$

where $A(y_i, \mathbf{x})$ and $I(y_i, y_j, \mathbf{x})$ stand for the association and iteration potentials, also named unary and pair-wise terms, respectively. The optimum class assignment $\hat{\mathbf{y}}$ given $\mathbf{x}$ is the one that maximizes the posterior, i.e.,

$$\hat{\mathbf{y}} = \arg\min_{\mathbf{y}} \left(\sum_{i \in S} A(y_i, \mathbf{x}) + \sum_{i \in S} \sum_{j \in N_i} I(y_i, y_j, \mathbf{x}) \right). \tag{4-2}$$

The unary term relates to the posterior probability that a pixel i takes a label y_i given the data $\mathbf{x}$. In this work, the posteriors are given by one of the FCNs described in the foregoing sections. Consequently, the unary of any pixel will consider a limited spatial context determined by the FCN largest receptive field. On the other hand, the pair-wise term expresses how labels at neighboring pixels, i and $j \in N_i$, interact given the observed data $\mathbf{x}$, where N_i is the neighborhood of pixel i. Notice that the pair-wise term allows for non-neighboring pixels to interact through a sequence of intermediate neighboring pixels. In this way, the CRF model is able to capture information of a context as large as the image itself.

Actually, using CRF inference as post-processing does not exploit the full potential of CRF. This is mainly because CNN is trained with no regard to the CRF post-processing. Nevertheless, it has been shown to be beneficial when combined with some of the aforementioned FCN architectures. However, this accuracy gain comes at the cost of increased computational complexity both for training and inference.

5
EXPERIMENTS AND RESULTS

In this chapter, we describe the set of experiments conducted to assess the performance of the different fully convolutional neural networks. The analysis is carried on a set of images captured by three different platforms.

The performance of the FCN designs is evaluated experimentally in terms of overall accuracy, F1-score, and Intersection Over Union. We also verify the benefits of conditional random fields (CRFs) as a post-processing step to improve the segmentation maps. Additionally, we compare visually the FCN outcome with the corresponding reference.

5.1
Study Area and Data Acquisition

Three different scenarios were chosen for evaluating experimentally the performance of the fully convolutional networks. First, using UAVs for the segmentation of single tree crowns in urban areas. Secondly, using aircraft images for individual building rooftop segmentation. Finally, using satellite images for the deforestation analysis in the Brazilian Amazon. The experiments on the three approaches are described in the following sections.

5.1.1
Single Tree Species - the cumbaru trees

The first scenario comprises UAV images of Campo Grande municipality, in the state of Mato Grosso do Sul, Brazil (Figure 5.1). This dataset was a subset of the one presented in [11] and comprises 225 UAV images acquired from 13 August 2018 to 22 September 2018 using a Phantom 4 advanced quadcopter (DJI Innovation Company Inc., China) in three study areas, depicted in Figure 5.1. The UAV was equipped with an RGB camera with 20 megapixels, a CMOS sensor, with a nominal focal length of 8.8 mm, and a field of view of 84°. The flight height ranged from 20 to 40 m over the targets, which assured a mean ground sample distance (GSD) of approximately 1 cm (Figure 5.2).

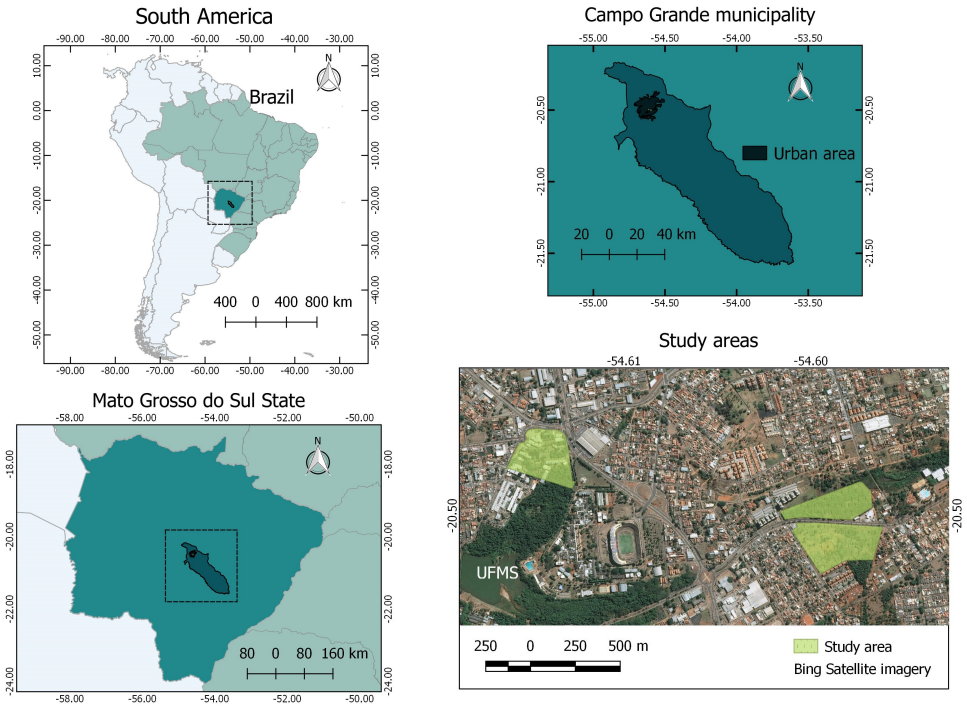


Figure 5.1: Study area at Campo Grande, Mato Grosso do Sul, Brazil.

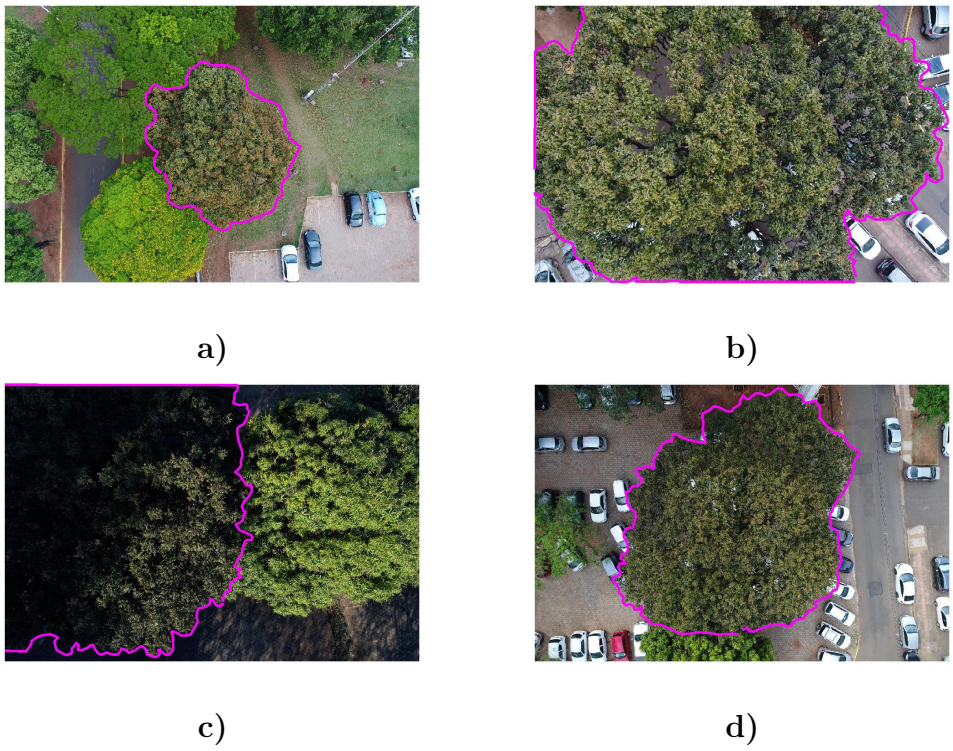


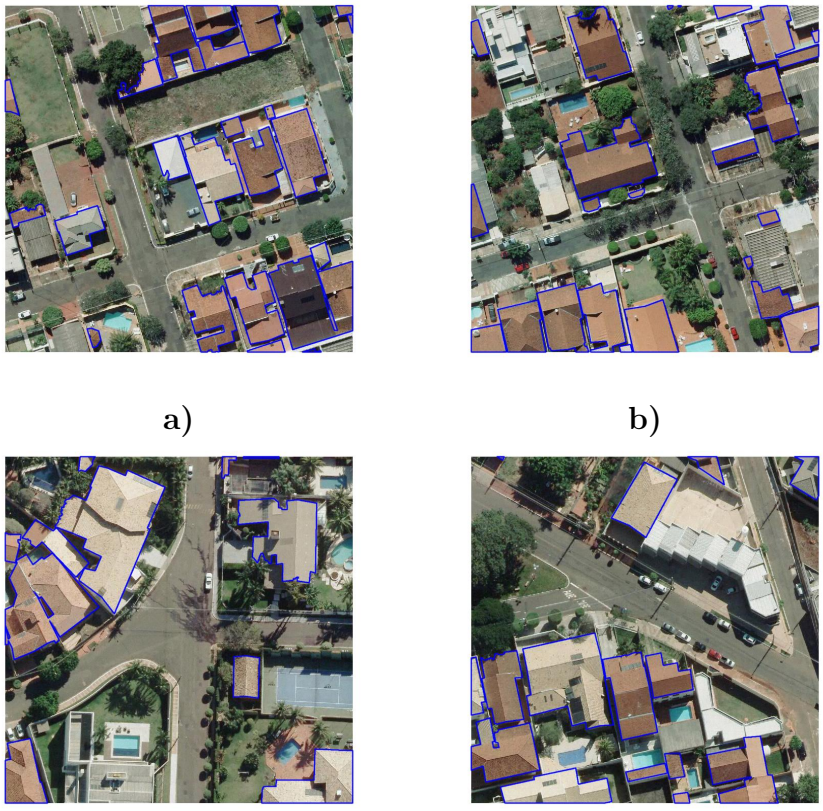
Figure 5.2: Image samples with the reference tree contour in pink, showing variations in terms of scale (**a**,**b**), illumination (**c**), and different urban patterns (**d**).

The target species is the cumbaru trees, an endangered and protected species in the target region. An analyst well acquainted with the target area produced the reference masks by delineating each single tree manually. Field inspections were performed to assure data quality.

The images were 5472 × 3648 pixels (20 megapixels) large and represent a wide range of appearances and scale variations. They were acquired at different times of the day and were therefore affected by different illumination conditions. The images were captured over diverse neighborhoods characterized by different urban patterns. The cumbaru class accounted for approximately 44% of the total pixels of the dataset.

5.1.2 Individual Building Rooftop

The dataset of the second study area was provided by Campo Grande city hall, a municipality in the state of Mato Grosso do Sul, Brazil (Figure 5.1). Also in this dataset, the images correspond to an urban scenario. The dataset comprises two aerial orthophotos with a spatial resolution of 5619×5946 pixels, and a GSD of 10 cm.



a) **b)**

c) **d)**

Figure 5.3: Image samples with the reference rooftop contour in blue.

The pixels corresponding to the rooftop class represents 23% of the total pixels in this dataset.

5.1.3 Brazilian Amazon Deforestation

The last study area corresponds to the portion of Amazon forest localized in the coordinates 09°36'51"S - 10°18'35"S latitude, and 062°56'41"W - 064°20'51"W longitude in the state of Rondônia, Brazil (Figure 5.4). The state extends over approximately 238.000 km^2, covering around 5% of the total Brazilian Amazon being one of the most deforested states. The area is characterized by dominant tropical swamp vegetation and a significant extension of wet and dry land savannas (termed "cerrados" in Brazil) [111].

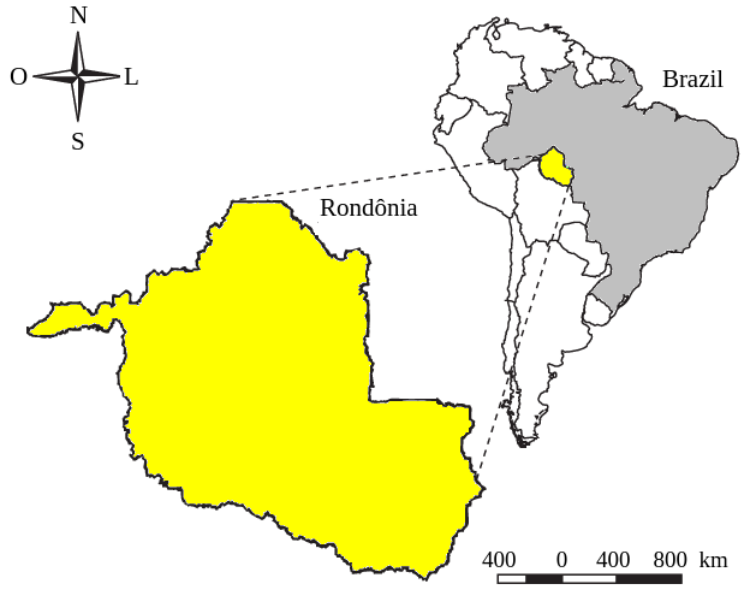


Figure 5.4: Study area at Rondônia, Brazil.

Data of deforestation in the Amazon have been made freely available on the website of the National Institute for Space Research (INPE) through two main initiatives: the DETER (Real-Time Deforestation Detection System) and PRODES (Amazon Deforestation Calculation Program) projects. In particular, the PRODES project monitors the Brazilian Amazon forest from images provided by the Landsat family with 30 meters of spatial resolution and 16-day temporal resolution. In our case study, we used the images from Landsat-8 Operational Land Image (OLI) with a spectral range of 7 bands and with a radiometric resolution of 16 bits.

The dataset comprises two Landsat-8 images of size 5120 × 2550, acquired on the period between July, 1^{st} of 2016 and August, 21^{st} of 2017,

searching for images with minimum cloud cover. The labeling of deforestation was done by visual photo-interpretation of images, carried out by qualified professionals. These experts identify the change patterns based on three main observable images elements: tone, texture, and context. Also, they only identify deforestation polygons with an area greater than 6.25 hectares.

For the creation of the reference maps, PRODES adopts a methodology of incremental mapping. In the production of incremental mapping, PRODES creates an exclusion mask (see Reference 1 in Figure 5.5), which covers the areas deforested in previous years, and another mask (see Reference 2 in Figure 5.5), which covers the deforestation on the reference year. The exclusion mask is used to eliminate the possibility of old deforestation being mapped again, so the work of photo-interpretation is done only on the reference year image with the new increments of deforestation.

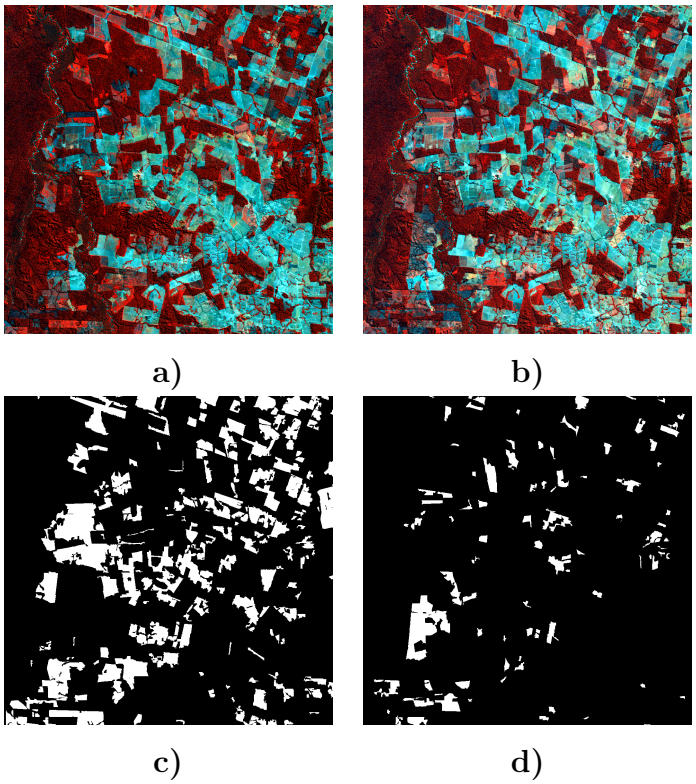


Figure 5.5: NIR-G-B composition of two crop of the images at dates 2016 and 2017 **a),b)**, respectively, **c)** Reference 1, and **d)** Reference 2

The dataset is highly unbalanced, with a representation of the class deforestation of only 2% of the total number of pixels in the image.

To reduce the classification errors at the borders between the prediction maps and the reference estimated visually by experts, we created a buffer inside the deforestation polygons. The buffer was created by the difference between

the dilation of four pixels and the erosion of two pixels of the deforestation polygons. These polygons were not considered for the training and test phases.

5.2 Evaluation Metrics

In the following experiments, Accuracy is reported in terms of three metrics: overall accuracy (OA), F1-score (F1), and intersection over union (IoU).

The overall accuracy is given by:

$$OA = \frac{tp + tn}{tp + tn + fp + fn} \tag{5-1}$$

where tp, tn, fp, fn stand for the number of true positives, true negatives, false positives and false negatives, respectively. In our analysis, positives and negatives refer to the pixels assigned by the underlying classifier. Such positives and negatives are true or false, depending on whether or not they agree with the ground truth, respectively.

The F1-score is defined as:

$$F1 = 2 \times \frac{P \times R}{P + R}, \tag{5-2}$$

where P and R stand for precision and recall, respectively, and are given by the ratios [112]:

$$P = \frac{tp}{tp + fp} \tag{5-3}$$

$$R = \frac{tp}{tp + fn} \tag{5-4}$$

For the semantic segmentation tasks, the intersection over union (IoU), also known as the Jaccard index, has often been used as an accuracy metric. IoU is given by the ratio of the number of pixels present both in the reference and in the prediction masks to the total number of pixels present across both masks [113], formally:

$$IoU = \frac{|Reference \cap Prediction|}{|Reference \cup Prediction|} \tag{5-5}$$

In addition, for the deforestation dataset in the Brazilian Amazon we also used the metric *Alarm Area (AA)* [49]. This metric is given by the ratio of the positives to the total samples in the test set.

$$AA = \frac{tp + fp}{tp + tn + fp + fn} \tag{5-6}$$

The metric represents the proportion of total area analyzed that the method considered to have been deforested. The usefulness of this metric

concerns an operational scenario in which the automatic method selects areas with a certain probability of having suffered deforestation, which will subsequently be visually evaluated by photo-interpreters. Therefore, the metric indicates the reduction of the analyst's effort in comparison to the analysis of the entire imaged area.

5.3 Network Architectures

We started our experiments with the networks' configurations exactly as defined in the corresponding original papers. Next, we varied some of their hyper-parameters, such as the number of layers, operations per layer, and the number and size of kernels, aiming to fine-tune each network to the target application. After these preliminary experiments, we selected for SegNet, U-Net, and FC-DenseNet the architectures described in Table 5.1 for cumbaru and rooftop segmentation. For both DeepLabv3+ variants, we adopted the original design as described in Sections 4.4 and 4.5. Table 5.2 shows for each network the total number of parameters that must be estimated by supervised training.

Table 5.1: Details of the SegNet, U-Net, and FC-DenseNet architectures used in the experimental analysis of Cumabru and Rooftop datasets

	SegNet		**U-Net**		**FC-DenseNet**	
	Layer	Kernel No.	Layer	Kernel No.	Layer	Kernel No.
Encoder	2 × SB + pool	32	UB + pool	32	conv1	32
	2 × SB + pool	64	UB + pool	64		
	3 × SB + pool	128	4 × (UB + pool)	128	8 × (DB + TD)	48
	3 × SB + pool	256	UB + pool	256		
	3 × SB + pool	512	UB	512	DB	176
Decoder	Up + 2 × SB	512	TC + Concat.+ UB	256		
	SB + Up	256	TC + Concat. + UB	128		
	2 × SB	256	TC + Concat. + UB	128		
	SB + Up	128	TC + Concat. + UB	128	8 × (TC + DB)	192
	2 × SB	128	TC + Concat. + UB	128		
	SB + Up	64	TC + Concat. + UB	64		
	SB	64	TC + Concat. + UB	32		
	SB + Up	32				
	SB	32				
	conv2 (1 × 1)	2	conv2 (1 × 1)	2	conv2 (1 × 1)	2
	Softmax		Softmax		Softmax	

For SegNet, we adopted the following notations: SB stands for the SegNet block (3 × 3 convolution + batch normalization + ReLU) and Up for unpooling layers. Concerning the U-Net, UB stands for U-Net block (2 × (3 × 3 convolution + ReLU)), and TC denotes a 3 × 3 transposed convolution. The FC-DenseNet was built from dense blocks (DB) of 2 layers, where each layer stands for (ReLU + 3 × 3 convolution). The transition down (TD) operation represents (ReLU + 1 × 1 convolution).

Table 5.2: Number of parameters of each network.

Method	**Parameters**
U-Net	11M
SegNet	16M
FC-DenseNet	0.4M
DeepLabv3+ (Xception)	41M
DeepLabv3+ (MobileNetV2)	2M

For the Amazon Deforestation, we ended up using a simpler network architecture as the total number of samples in this dataset was significantly fewer than the other two applications. In this way, we avoided over parametrized networks which can lead to over-fitting and lower generalization performance. We conducted our experiment with the configuration for Segnet, U-Net, and FC-DenseNet described in Table 5.3.

Table 5.3: Details of the SegNet, U-Net, and FC-DenseNet architectures used in the experimental analysis of Amazon Deforestation.

	SegNet		**U-Net**		**FC-DenseNet**	
	Layer	Kernel No.	Layer	Kernel No.	Layer	Kernel No.
Encoder	2 × SB + pool	32	UB + pool	32	conv1	32
	2 × SB + pool	64	UB + pool	64		
	3 × SB + pool	128	UB + pool	128	8 × (DB + TD)	64
			UB	128		
Decoder	Up + 2 × SB	128	TC + Concat.+ UB	128		
	SB + Up	64	TC + Concat. + UB	64		
	SB	64	TC + Concat. + UB	32		
	SB + Up	32			8 × (TC + DB)	128
	SB	32				
	conv2 (1 × 1)	2	conv2 (1 × 1)	2	conv2 (1 × 1)	2
	Softmax		Softmax		Softmax	

For SegNet, we adopted the following notations: SB stands for the SegNet block (3 × 3 convolution + batch normalization + ReLU) and Up for unpooling layers. Concerning the U-Net, UB stands for U-Net block (2 × (3 × 3 convolution + ReLU)), and TC denotes a 3 × 3 transposed convolution. The FC-DenseNet was built from dense blocks (DB) of 4 layers, where each layer stands for (ReLU + 3 × 3 convolution). The transition down (TD) operation represents (ReLU + 1 × 1 convolution).

For the DeepLabv3+ architecture, we reduced the number of layers in the encoder path for both variants. The decoder structure remained the same as described in section 4.4 for the Xception and MobileNetV2 backbones. The encoder for both variants followed the structure illustrated in Table 5.4.

Table 5.4: Details of the encoder architectures of DeepLabv3+ variants used in the experimental analysis of Deforestation.

	Xception				Mobilenetv2			
	Layer	Kernel No.	Times	Stride	Layer	Kernel No.	Times	Stride
Encoder	conv1(3×3)	32	1	2	conv1(3×3)	32	1	2
	conv1(3×3)*	64	1	1				
	SC	128	2	1	IRB	16	1	1
	SC-last	128	1	2	IRB	24	2	2
	Shortcut	128	1	2	IRB	32	3	2
	Add(Shortcut, SC-last)	128	1	2				
	SC	256	2	1				
	SC-last	256	1	2				
	Shortcut	256	1	2				
	Add(Shortcut, SC-last)	256	1	1				

For the Xception model, we adopted the following configuration: SC stands for Separable Convolution block (Depthwise Convolution + Batch-Normalization + (1×1) Convolution + Batch-normalization), Shortcut to indicate the residual (1×1) convolution, and * to highlight the input feature map of the shortcut connection. Mobilenetv2 was built from inverted residual block (IRB), where each block stands for a ((1×1) Convolution + Depthwise Convolution + (1×1) Convolution), as described in Section 4.5.

Table 5.5: Number of parameters of each network for the Deforestation application.

Method	Parameters
U-Net	1.4M
SegNet	0.88M
FC-DenseNet	0.29M
DeepLabv3+ (Xception)	1.1M
DeepLabv3+ (MobileNetV2)	0.21M

5.4 Experimental Setup

The networks were implemented using the Keras deep learning framework [114] on a system with the following configuration: Intel(R) Core(TM) i7 processor, 64 GB of RAM, and NVIDIA GeForce GTX 1080Ti GPU. All models in the three applications were trained from scratch for up to 100 epochs using the Adam optimizer. The decay of the first and second moments was set as described in [115]. Early stopping was used to avoid over-fitting. Training stopped when the performance in the validation set degraded over ten consecutive epochs. In the end, the model that exhibited the best performance in the validation set across all executed epochs was kept for the test phase.

To evaluate the generalization of the network architectures in the cumbaru and rooftop applications, we applied five-fold cross-validation for each method. Thus, for each fold, the total dataset was randomly split into three disjoint sets: 70% for training, 10% for validation, and 20% for testing.

In particular, for cumbaru segmentation, the images were divided into non-overlapping patches of 512×512 pixels for all models, for a total of 5250 patches in the entire dataset. The final segmentation of the entire image was the mosaic of all patch-wise segmentation outcomes.

For rooftop segmentation, the two orthophotos were divided into 49 non-overlapping tiles of size 1024×1024. Similar to cumbaru segmentation, the images were split in patches of 512×512 pixels with 50% overlap. Specifically, 196 patches were cropped for the total dataset.

For these two datasets, we empirically adjusted the batch size for each model individually, also taking into account the GPU memory demand and availability. The batch size was set to 16 for the U-Net, 8 for the SegNet, 6 for the FC-DenseNet, 2 for the DeepLabv3+ with the Xception backbone, and 6 for the DeepLabv3+ with MobileNetV2. We tested Deeplabv3+ with an output stride equal to 16 for a field of view of 32×32. The selected atrous rate was 6, 12, and 18, as proposed in [98]. The hyperparameter learning rate was set to 10^{-4}. We used for both applications the binary cross-entropy loss.

For the Deforestation dataset, we selected a combination of pairs of co-registered Landsat images, acquired on the dates 2016 and 2017, as specified in section 5.1.3. We combined the images of both date following the Early Fusion method [49].

Each Landsat image comprised the Coastal Aerosol, Blue, Green, Red, near-infrared regions (NIR), Short Wave Infrared 1 and 2 (SWIR1, SWIR2 respectively) for a total of 7 spectral bands. We also included the Normalized difference vegetation index (NVDI), from Red (visible) and near-infrared bands (NIR), as shown in Equation 5-7. We also added the NDVI as a 8th band.

$$NDVI = \frac{NIR - Red}{NIR + Red} \tag{5-7}$$

The resultant NDVI values were concatenated along the spectral dimension for a total of 8 bands per each Landsat image.

We divided the images into 100 non-overlapping tiles of size 250×512. Therefore, we split the total set of tiles in 20% for training, 5% for validation, and 75% for testing.

The extracted input tiles are split into patches equal to 128×128 with an overlap of 80%, for a total of 19503 in the entire image.

Taking into account the proportion of deforestation samples (about 2%) on the image with respect to the non-deforestation class (around 98%), we applied data augmentation only considering patches with the presence of deforestation areas. In this way, we tried to bring some balance by increasing the number of samples of the under-represented class. These samples were augmented by applying 90° rotations, horizontal and vertical flip transformations for the training patches.

We also applied weighted cross-entropy loss, as seen in equation 5-8. The idea is to assign a larger weight to the weakly represented class.

$$Weighted\ Loss = -\frac{1}{NM}\sum_{i=1}^{N}\sum_{j=1}^{M}[w_d y_{(i,j)} log(\hat{y}_{(i,j)}) + w_{nd}(1 - y_{(i,j)}) log(1 - \hat{y}_{(i,j)})] \tag{5-8}$$

where N stands for the total number of training pixels, w_d and w_{nd} for the weights of the *deforestation* class and *non-deforestation*, respectively. Moreover, y_i and $\hat{y}_i$ represent the target and predicted label at pixel i.

The weights for the classes *deforestation* and *non deforestation* were empirically set to 2 and 0.4, respectively.

The batch-size configuration in this application was selected experimentally to 16 for all tested networks.

5.5 Post-processing CRF

As for the post-processing, we adopted a fully connected CRF, considering that all pixels were connected to all other pixels in the image, Equation (4-2) took the form:

$$\hat{\mathbf{y}} = \arg\min_{\mathbf{y}} \left(\sum_{i \in S} A(y_i, \mathbf{x}) + \sum_{i,j \in S} I(y_i, y_j, \mathbf{x}) \right) \tag{5-9}$$

We defined as association potential $A(y_i, \mathbf{x}) = -\log P(y_i|\mathbf{x}_i)$, where $P(y_i|\mathbf{x}_i)$ denotes the posterior probability given the FCNs tested in this work. As in [15] and [110], we used for the pair-wise term the following expression:

$$\begin{aligned} I(y_i, y_j, \mathbf{x}) = \mu(y_i, y_j) \Bigg[& w_1 \exp\left(-\frac{||\mathbf{c}_i - \mathbf{c}_j||^2}{2{\sigma_\alpha}^2} - \frac{||\mathbf{x}_i - \mathbf{x}_j||^2}{2{\sigma_\beta}^2}\right) \\ & + w_2 \exp\left(-\frac{||\mathbf{c}_i - \mathbf{c}_j||^2}{2{\sigma_\gamma}^2}\right) \Bigg] \end{aligned} \tag{5-10}$$

where $\mu(y_i, y_j) = 1$ if $y_i \neq y_j$, and zero otherwise, $\mathbf{x}_{i,j}$ represents the observed data at pixel i, j, and $\mathbf{c}_{i,j}$ denotes the pixel spatial coordinates. The hyperparameters for the cumbaru application w_1, w_2, σ_α,

σ_β and σ_γ were set to 1, 1, 80, 13, and 3, respectively, which corresponded to their default values as proposed in [110]. In the case of the rooftop segmentation, the hyperparameters σ_α, σ_β which controls the proximity and similarity between pixels were set to 1. For our experiments, we adapted the fully CRF code available at `https://github.com/Golbstein/Keras-segmentation-deeplab-v3.1/blob/master/utils.py`.

5.6 Results

In this section, we present the results of the experimental evaluation of the selected semantic segmentation approaches, as well as a visual analysis of the segmentation outcomes. The experiments were carried out on images obtained from different platforms, UAVs for single cumbaru segmentation, aerial images for rooftop pixel-wise classification, and satellite images for Deforestation in Amazon forest.

5.6.1 Performance Evaluation for Cumbaru Segmentation

5.6.1.1 Segmentation Accuracy for Cumbaru Segmentation

Segmentation Accuracy Figure 5.6 shows the average results over five fold cross-validation for each method. All networks performed well in the task, achieving OA and an F1-score above 85% and IoU above 75%. The plot also shows pictorially the standard deviation for each metric across the folds. The standard deviation was about 1% for OA and a little larger for F1 and IoU.

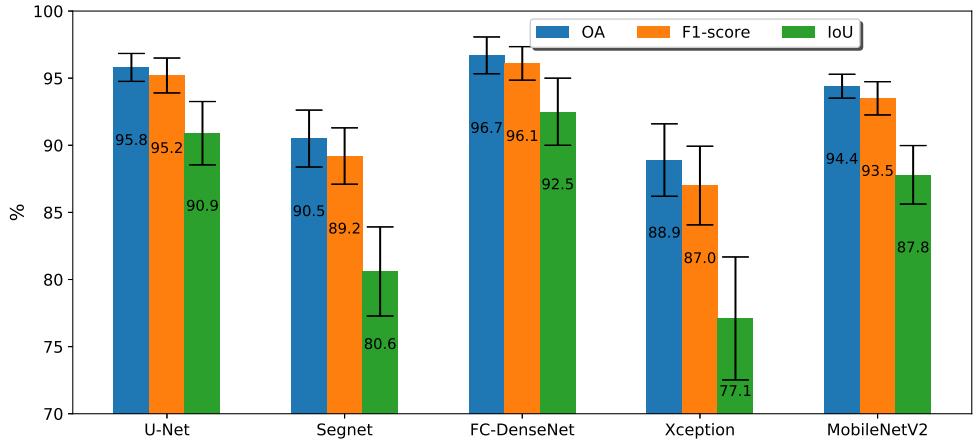


Figure 5.6: Mean of the overall accuracy (OA), F1, and IoU in the fivefold cross-validation for the all FCN architectures.

FC-DenseNet was the most accurate among the tested architectures. It reached on average OA = 96.7%, F1 = 96.1%, and IoU = 92.5%. It outperformed the second ranked network, the U-Net, in 0.9% in terms of OA and F1 respectively, and 1.6% for IoU.

Shortly behind U-Net came DeepLabv3+ in the MobileNetV2 version. The differences between these two architectures in all three metrics were about 1.4%, 1.7%, and 3.1% in terms of OA, F1, and IoU, respectively. Figure 5.6 also shows the standard deviation around the mean values recorded by each architecture along the five folds for all three metrics. These three high ranked methods also presented lower dispersion than the other two in our experiments, behaving fairly stable across the folds.

The low variation across the folds observed in our experiments is an indication of the better generalization ability of these three methods.

SegNet's architecture ranked forth, staying 3.9%, 4.3%, and 7.2% behind DeepLabv3+ MobileNetV2, in terms of OA, F1, and IoU, respectively. The range within which performance varied in our experiments confirmed that SegNet stood behind the first three architectures in the ranking. The inferior SegNet's results were most probabily due to the way it recoveres the original image resolution in the network expansion stage. SegNet employed interpolation, while U-Net and FC-DenseNet used transposed convolution. Moreover, the skip connections of U-Net and FC-DenseNet were more effective in recovering high resolution spatial details than the consideration of pool indices by SegNet.

In the recent few years, the DeepLabv3+ Xception has been regarded as the state-of-the art in semantic segmentation. Nevertheless, it achieved in our experiments the worst performance among all tested architectures, both in terms of absolute average accuracies and in terms of variability across the five folds.

The DeepLabv3+ variants brought no improvement in relation to U-Net and FC-DenseNet results. This suggests that the larger receptive fields induced by the dilated convolutions was little relevant in this particular application.

Figure 5.7 shows the performance after post-processing the results produced by each network with a fully connected CRF. Compared with the results of Figure 5.6, CRF brought just a slight improvement for all network designs. The profile in Figure 5.7 is quite similar to that of Figure 5.6. Again, after CRF post-processing, FC-DenseNet was the best performing architecture, followed by U-Net, DeepLabv3+ MobileNetV2, and then SegNet with DeepLav3+ Xception as the worst performing network.

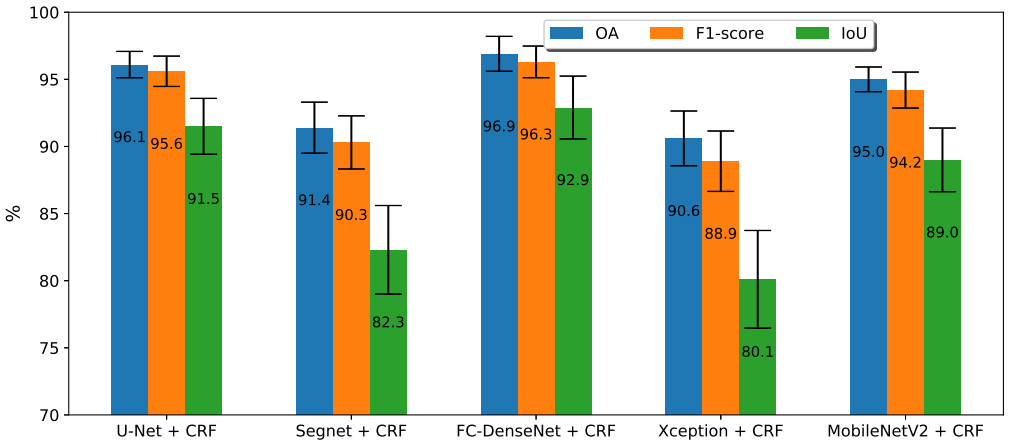


Figure 5.7: Average of the accuracy, F1-score, and IoU in fivefold cross-validation for all the methods using CRF.

To better visualize the benefits of post-processing, Figure 5.8 shows just the accuracy gain brought by CRF. DeepLav3+ with the Xception backbone was the network that most profited from CRF. In second and third place stand SegNet and DeepLabv3+ MobileNetV2, respectively. The gains for U-Net and FC-DenseNet were considerably lower, below 0.6%.

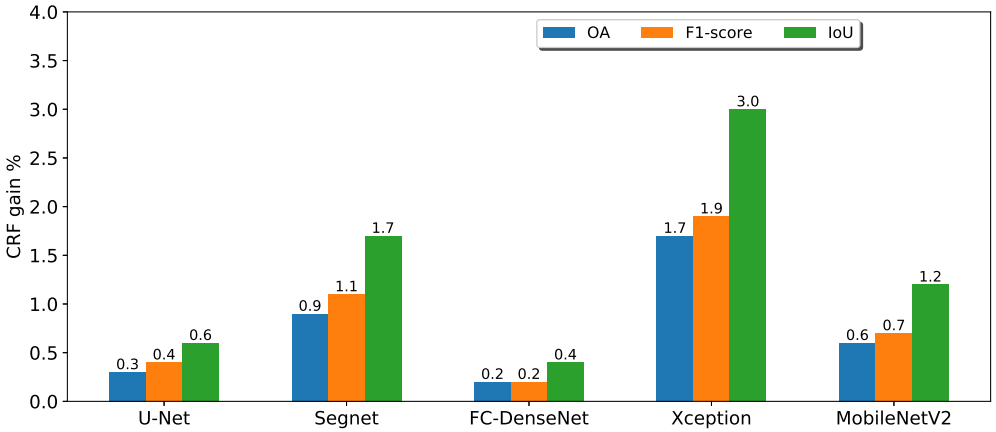


Figure 5.8: Performance gains due to CRF in terms of overall accuracy, F1-score, and IoU.

5.6.1.2 Visual Analysis for Cumbaru Segmentation

Figures 5.9–5.13 show the outcomes of all methods for five sample images of our dataset. References are shown on the left as the ground truth mask

overlaid on the input image. The next four columns on the right contain the segmentation produced by each network, whereby the upper and lower rows correspond to the outcome prior to and after CRF post-processing, respectively.

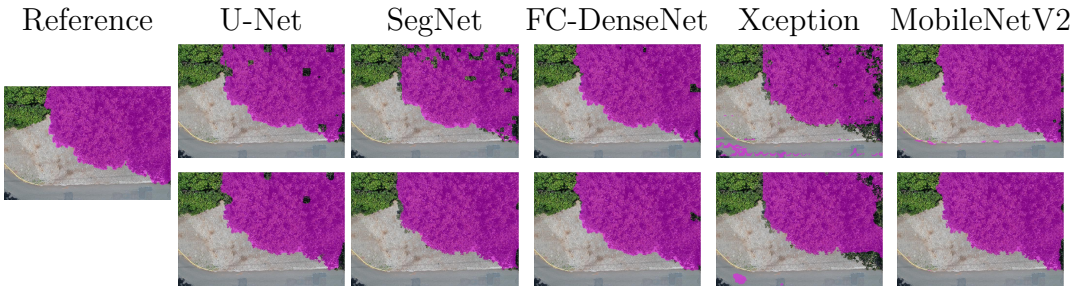


Figure 5.9: Sample Segmentation 1 prior (first row) and after CRF post-processing (second row).

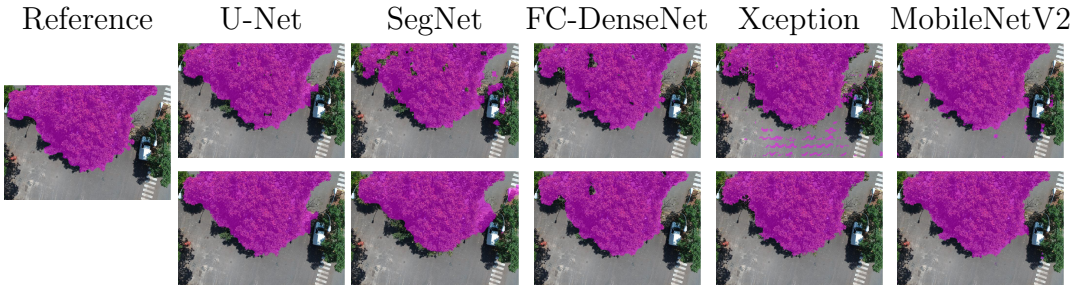


Figure 5.10: Sample Segmentation 2 prior (first row) and after CRF post-processing (second row).

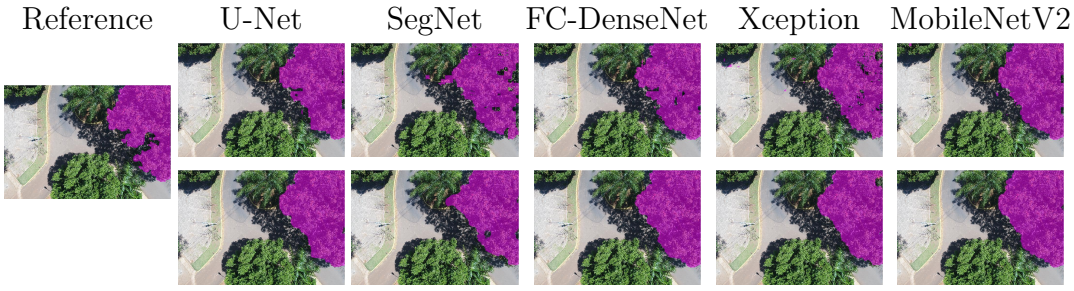


Figure 5.11: Sample Segmentation 3 prior (first row) and after CRF post-processing (second row).

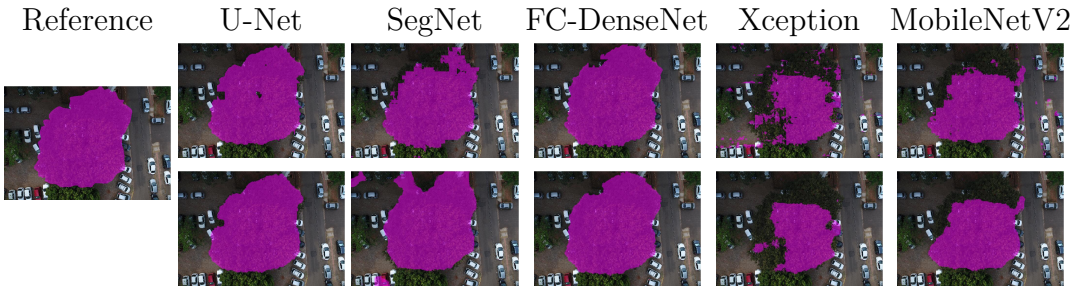


Figure 5.12: Sample Segmentation 4 prior (first row) and after CRF post-processing (second row).

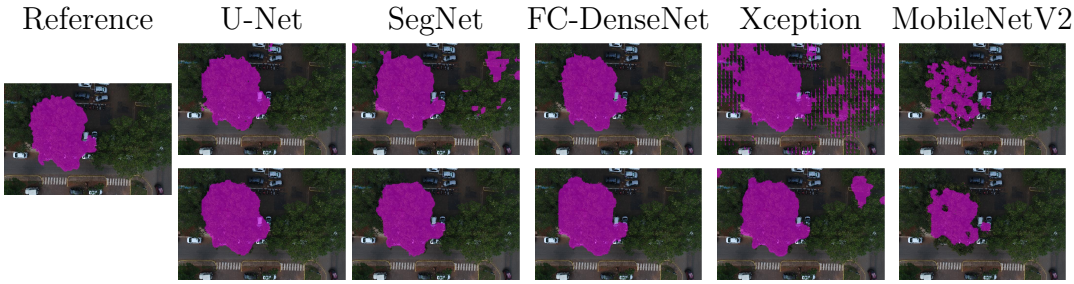


Figure 5.13: Sample Segmentation 5 prior (first row) and after CRF post-processing (second row).

We first analyzed the results prior to CRF post-processing. More than the U-Net architecture, SegNet tended to produce holes in the canopy region. This effect is especially apparent in Figures from 5.9 to 5.12. Figures 5.10 and 5.13 show that SegNet also produced a number of false positives in some images. Often, it also failed to detect the canopy edges, as exemplified in Figure 5.12.

The Figures show that U-Net outperformed SegNet in virtually all test images. Actually, the MobileNetV2 version of DeepLabv3+ generated fewer holes than U-net in some images. However, DeepLabv3+ MobileNetV2 was more prone to produce false positives (see Figures 5.9, 5.10, and 5.12).

Both DeepLabv3+ variants often failed over dark canopy regions as exemplified by Figures 5.12 and 5.13. These figures also show that SegNet performed better than the DeepLabv3+ variants over dark areas, but not as well as the other networks. FC-DenseNet and U-Net's results contained fewer false positives and false negatives in dark canopy regions.

The figures also reveal that the Xception variant of DeepLabv3+ achieved the poorest performance among the evaluated architectures, due to holes (Figures 5.9 and 5.11), false positives (Figures 5.9, 5.10, and 5.12), and poorly classified dark areas (Figures 5.12 and 5.13).

The effects of CRF post-processing on the segmentation outcome can be observed by comparing the results shown in the upper and lower rows of Figure 5.9 to Figure 5.13. CRF acted by smoothing the class labels produced by the networks. It was particularly effective at filling holes and suppressing small regions of false positives. That was related to the fact that both DeepLabv3+ variants profited more than all other networks from CRF post-processing as indicated in Figure 5.8. However, these errors corrected by CRF came about as few small regions that represented in terms of the number of pixels a small proportion of the entire image. Thus, although CRF contributed to producing cleaner segmentation outcomes, such improvements did not manifest in a significant change in the accuracy metrics. This could be observed even in the results of FC-DenseNet, the best performing network among all tested approaches. Note in Figures 5.9 to 5.12 that CRF managed to remove small holes left by FC-DenseNet in the canopy region.

5.6.2 Performance Evaluation for the Rooftop Segmentation

5.6.2.1 Segmentation Accuracy for Rooftop Segmentation

The bar graph in Figure 5.14 summarizes the results of the experiments carried out in the Rooftop dataset. The figure shows the classification performance achieved by the FCNs approaches in terms of Overall Accuracy (OA), F-1 score, and IoU over 5-fold cross-validation.

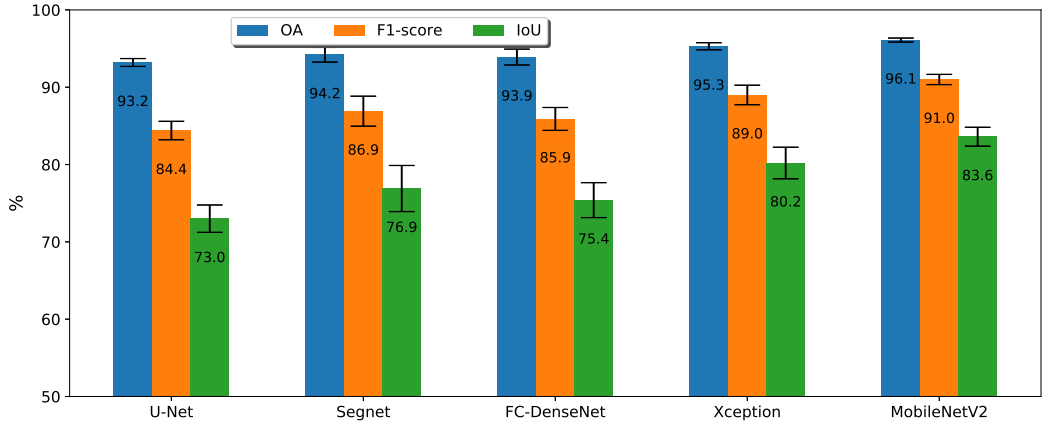


Figure 5.14: Mean of the overall accuracy (OA), F1, and IoU in the fivefold cross-validation for the all FCN architectures.

The first three bar groups correspond to the U-Net, SegNet, and FC-DenseNet methods. Segnet and FC-DenseNet performed similarly with a slight

difference of 0.3%, 1% and 1.5% in terms of Overall Accuracy, F-1 score, IoU, respectively, over the five-folds. As illustrated in the figure, the results of both methods were superior to U-Net, with a difference of 1% for OA, 2.5% for F1-score, and 3.9% for IoU in relation to SegNet. This better results achieved by SegNet and FC-DenseNet might be attributed to the use of the batch-normalization layers that deal with the internal covariate shift problem.

The last group of bars corresponds to the DeepLabv3+ variants. In this case, DeepLabv3+ MobileNetV2 overcame the results of the Xception version with a difference of about 0.8% for OA, 2% for F1-score, and 3.4% for IoU. Figure 5.14 also reveals that MobileNetV2 and Xception variants consistently achieved better performance on the test set among all the FCN models. This superiority of the DeepLabv3+ variants might have been caused by the atrous convolution, which effectively expands the field of view of the filters to capture more contextual information. This effect is quite remarkable on this dataset given the presence of roofs with different sizes and other objects with low interclass variance in each image patch.

Figure 5.15 refers to the FCNs approaches using the CRF post-processing.

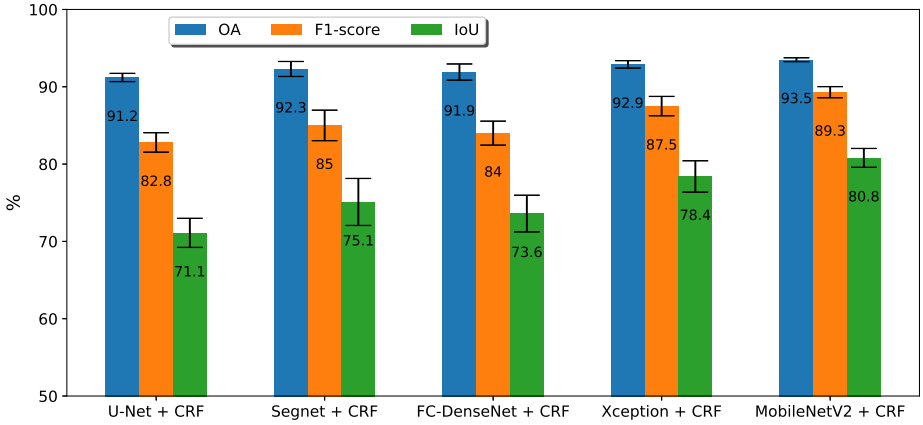


Figure 5.15: Mean of the accuracy, F1-score, and IoU in fivefold cross-validation for all the methods using CRF.

The CRF had a negative effect on the accuracy when compared with the results in Figure 5.14. Being more specific, the CRF reduced the accuracy in almost 2% for OA, F1-score, and IoU for all variants. The results after the CRF step present a similar profile as in Figure 5.14. Again, the DeepLabv3+ variants achieved the best results followed by SegNet and FC-DenseNet with a similar but slightly better performance than the U-Net model.

5.6.2.2 Visual Analysis for Rooftop Segmentation

In addition to the quantitative results, we further performed a qualitative analysis of the networks' outcomes. Figures 5.16 to 5.20 present five outcome instances, the reference and the segmentation before and after applying the CRF. The first row of the Figures contain the prediction maps of each of the tested FCNs approaches, and the second row shows the corresponding outcomes after the CRF post-processing.

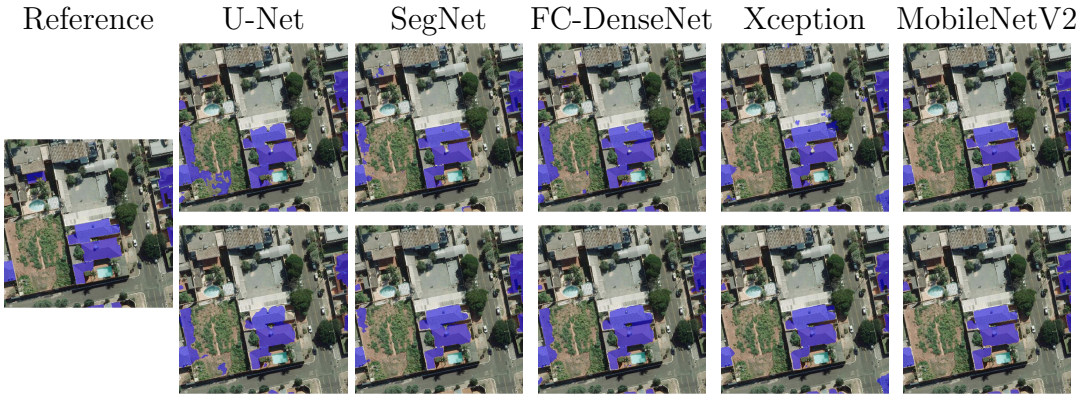


Figure 5.16: Sample Segmentation 5 prior (first row) and after CRF post-processing (second row).

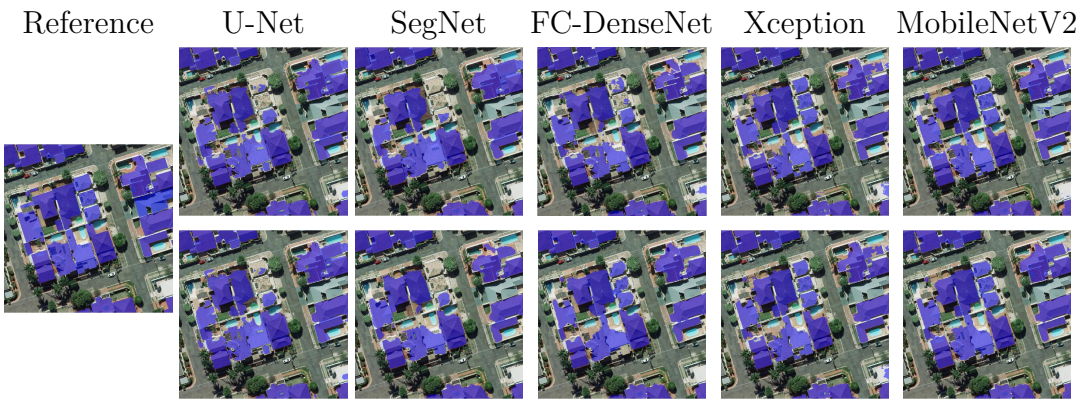


Figure 5.17: Sample Segmentation 5 prior (first row) and after CRF post-processing (second row).

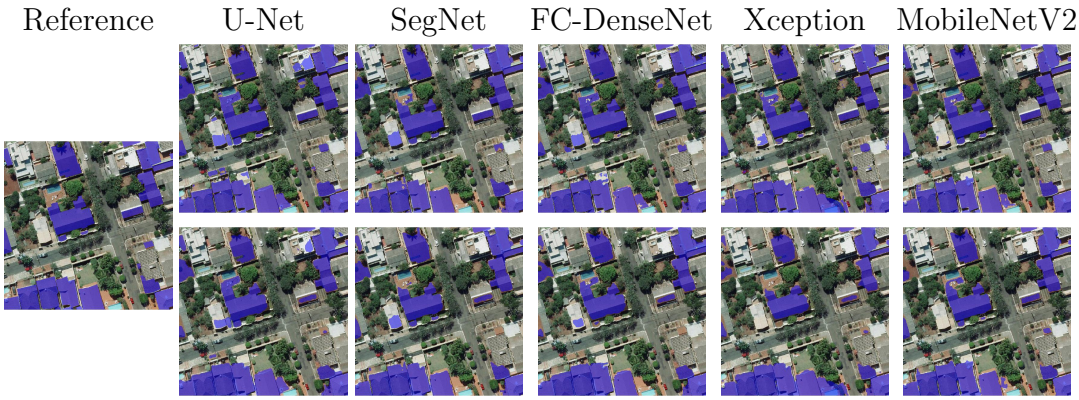


Figure 5.18: Sample Segmentation 5 prior (first row) and after CRF post-processing (second row).

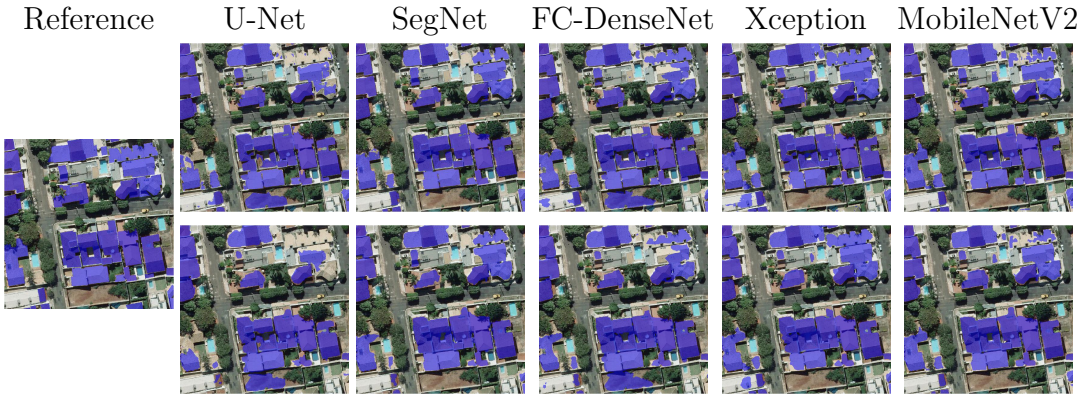


Figure 5.19: Sample Segmentation 5 prior (first row) and after CRF post-processing (second row).

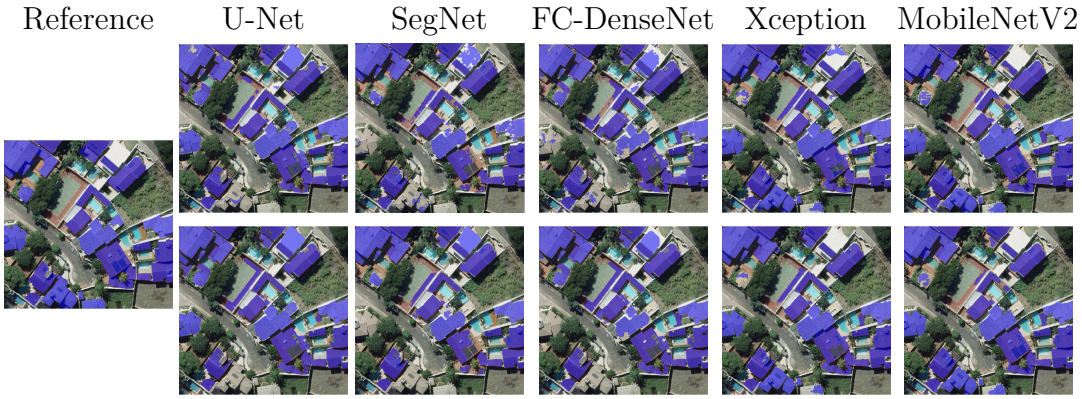


Figure 5.20: Sample Segmentation 5 prior (first row) and after CRF post-processing (second row).

A comparison of the prediction maps before the CRF step with the reference confirms the results shown in Section 5.6.2.1. The DeepLabv3+

networks managed to correctly classify most of the samples, generating less false positives and false negatives than the other FCNs architectures. This is especially apparent in figures 5.16, 5.19, and 5.20. In particular, it can be noted that MobileNetV2 more accurately delimited the edges of the rooftops.

In contrast with the Deeplabv3+ variants, it is specially seeing in Figures 5.17 and 5.20 that the U-Net, Segnet and FC-DenseNet methods were not able to identify some roofs. Consider for example in Figure 5.20 the roofs not recognized at the bottom of the image. This might have been caused because some roofs look like the street. We believe that the ability of DeepLabv3+ variants to consider context at multiple scales has helped to recognize these rooftops.

The second rows of Figures 5.17 and 5.20 correspond to the prediction maps after the CRF step. Comparing the results before and after CRF, we observe that this post-processing refined the segmentation results in some cases, reducing the number of false positives and false negatives, as is particularly apparent in Figure 5.16. However, the CRF step tended to eliminate some small roofs predictions, as seen in Figure 5.18. Also, the CRF merged roofs close to each other. This effect was particularly apparent in Figures 5.18 and 5.19, respectively.

5.6.3 Performance Evaluation for the Deforestation Detection

5.6.3.1 Segmentation Accuracy for Deforestation Detection

Figures 5.21 presents the accuracy results obtained by the FCNs for the Amazon dataset in terms of Overall Accuracy, Recall, Precision, and F1-score. The figure summarizes the results in terms of mean values and standard deviations computed after 50 runs. Each run with the same configuration of training and test samples as described in Section 5.4.

We carried out no experiments with the CRF with this dataset given the few samples that represent a deforestation region. As CRF encourages label agreement between neighboring pixels, this effect may harm the quality of final segmentation by removing most of these small regions.

For this dataset, all networks achieved similar Overall Accuracies above 98%. This is not surprising, as the dataset is dominated by one single class, the non deforestation class.

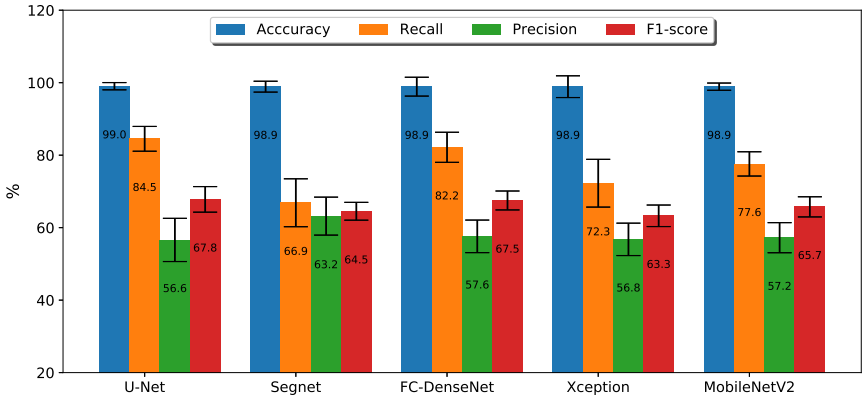


Figure 5.21: Mean of the overall accuracy (OA), Recall, Precision, and F1, over 50 run for the all FCN architectures.

In terms of F1-score, U-Net and FC-DenseNet achieved the best results among all FCN approaches, with a slight difference between them of 0.3%. DeepLabv3+ MobileNetV2 architecture ranked third, with a better performance than Segnet and DeepLabv3+ Xception, and behind U-Net and FC-DenseNet. As shown in the Figure, MobileNetV2 showed comparable results with the U-Net and FC-DenseNet in terms of precision, but, with a difference in terms of recall of about 6.9% and 4.6%, respectively, that degraded the F1-scores. On the other hand, Segnet and Xception models showed results inferior to MobileNetV2 of about 1.2% and 2.4%,respectively.

Figure 5.22 shows the amount of False Positive and False Negative for each network. Segnet produced the smallest number of incorrectly classified pixels and produced the least false positives. Regarding the number of false-negatives, it was clearly behind the other approaches. The comparatively low precision of U-Net, FC-DenseNet, Xception, and MobileNetV2 is related to the high number of false-positives yielded by these networks. In terms of false-negatives, these methods showed a clear superiority over Segnet.

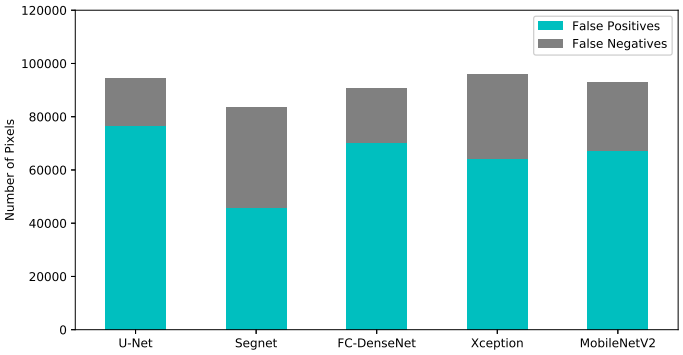


Figure 5.22: Mean Pixel Distributions

In Figure 5.23, we evaluate the method in terms of the Alarm Area and Recall defined in Section 5.2. The curve is constructed by using a thresholding method, where each point of the image represents the decrease from one to zero of this threshold with steps of 0.001. In terms of Overall Accuracy, a high threshold decreases the misclassifying errors of the FCNs due to the over-estimation of deforestation samples, highlighting those that should deserve more attention. A photo-interpreter then visually analyzes the image, or an inspector could be sent to the indicated areas to check what was real deforestation and what was just a false alarm. This reduces the human effort of the visual interpretation task; however, it also increases the error due to the under-estimation of the positive samples. These factors could be addressed by finding a balance between the number of potentially deforested samples indicated by the Alarm Area and the quantification of the avoidance of false-negatives samples (or Recall), as the probability threshold varies. In this sense, it is desirable in the scheme a high value of Recall with a low value of the Alarm Area.

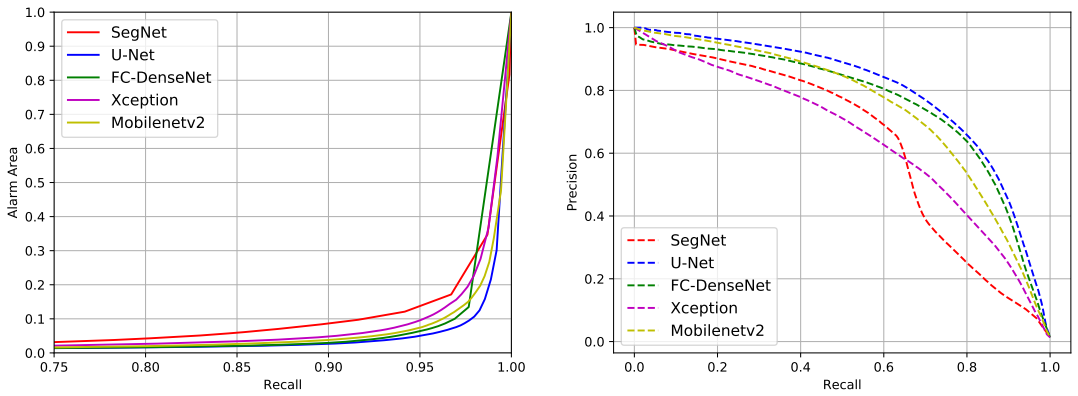


Figure 5.23: Evaluation performance of the deforestation approach considering Alarm Area and Precision versus Recall.

As noticed in Figure 5.23a), all methods achieved Recall values of about 90% when looking at less than 10% of the whole imaged area. It means that 90% of the correctly identified deforestation is contained in 10% of the image. Hence, instead of looking at the entire image, the analyst would focus on 10% of it, reducing human work by 90%. Beyond this value (90%), U-Net presented the best performance. It managed to classify more deforested samples correctly, with a minimum increase in the area to be observed. As expected, for threshold values close to 0, Recall and Alarm Area showed the highest performance, about 100%, since most of the samples were classified as deforestation.

The Precision and Recall curves are also summarized in Figure 5.23b). The main benefit of this scheme is to reduce the high cost that implies the displacement to some difficult access Amazon regions suspected of deforestation. In this scheme, it is a desirable outcome a high threshold with high precision values. Considering the area under the curve, the U-Net describes the best system with the highest precision as the thresholds varies.

5.6.3.2 Visual Analysis for Deforestation Detection

Figure 5.24 shows the results of Deforestation detection in two adjacent tiles of the test set. Figures 5.24 a) and b) show the images in 2016 and 2017, respectively, followed a NIR-G-B composition (Near Infrared, Green, and Blue bands). Figure 5.24 c) corresponds to the reference map, where unchanged and changed classes are colored in blue and green, respectively. Figures 5.24 d)-i) stand for the prediction maps produced by each fully convolutional network.

Compared with the reference map, the models detected roughly the same deforestation sites, and corroborate the results presented in section 5.6.3.1. The deforestation results of U-Net and FC-DenseNet methods are more consistent with the reference maps, as shown in Figures 5.24 d) and f). Next, we have the MobileNetV2, showing comparable results due to more false-positives and false-negatives samples. Segnet and Xception presented a comparatively higher tendency to produce false-negatives.

All FCN models were able to accurate classify the deforestation patches, although, U-Net and FC-DenseNet performed a bit better. A design feature that distinguishes these networks from their counterparts is the use of skip connections. However, the performed experiments do not allow attributing the better accuracies achieved by these networks to this feature.

5.6.4 Computational Complexity

In this section, we compare the methods in terms of computational load for training and inference. Table 5.6 and 5.7 presents the average training and inference times measured on the hardware infrastructure described in Section 5.4. The training time represents the mean time among the five folds for each method for the Rooftop and Cumbaru datasets. For the Deforestation detection, the training time stands for the mean time for the 50 runs. The mean inference time stands for the average time taken by each model to make a prediction for a whole image. All the models were trained from scratch with the same optimizer and learning rates for all the applications.

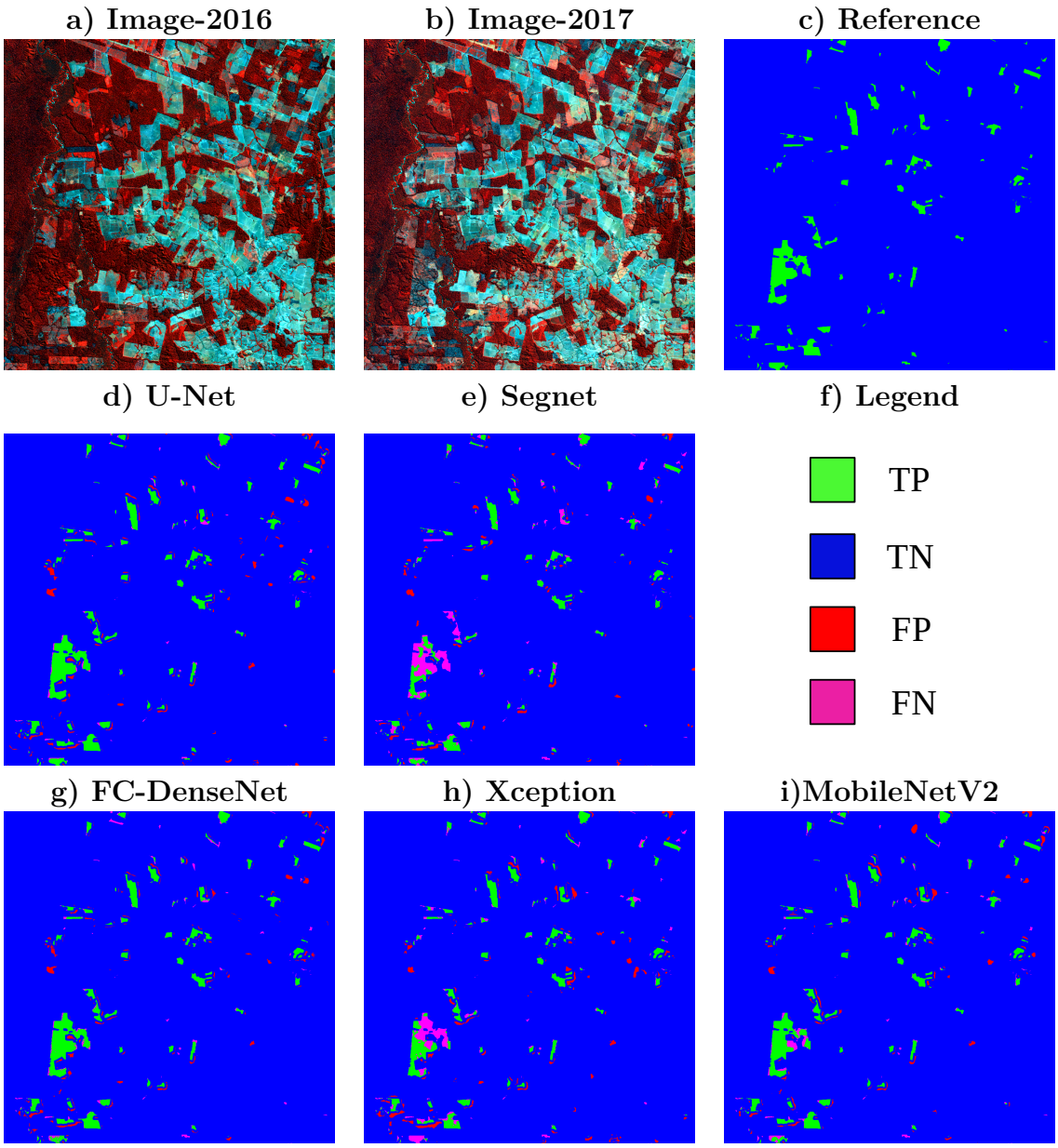


Figure 5.24: Prediction maps produced by U-Net, Segnet, FC-DenseNet, Xception and MobileNetV2 in two adjacent tiles of the test set.

Table 5.6: Average processing time for each method for Cumbaru and Rooftop dataset.

Method	Cumbaru dataset		Rooftop dataset	
	Training Time (h:min)	Inference Time (s)	Training Time (h:min)	Inference Time (ms)
U-Net	13:15	1.15	0:47	250
SegNet	09:12	1.17	1:11	350
FC-DenseNet	15:03	1.14	0:31	220
DeepLabv3+ (Xception)	20:33	4.44	0:31	870
DeepLabv3+ (MobileNetV2)	10:46	2.26	0:17	540
CRF	-	35.17	-	8.35

The cumbaru dataset contains more high resolution images than the Rooftop dataset. In consequence, it took longer for training and inference.

The difference in training times for each of the methods is given by the number of iterations or epochs the FCNs needs to converge, as monitored by the Early Stopping. On the other hand, additional layers in a deep network come at cost of a higher computational load in the inference. As networks continue to get deeper, these costs may become prohibitive for real-time applications. In particular, the DeepLabv3+ variants were deeper than the other three methods, which implied longer inference times in both applications. Finally, the CRF post-processing implied additional execution times, which prevents its usage in applications where the processing time is crucial.

Table 5.7 reports the average of the training and prediction time in the test set for each run on the Deforestation dataset. The table exhibits the lowest training times when compared with the other two applications, because this dataset is composed of a single image. Besides, the architecture is the simplest one among the tested counterparts. It should be mentioned that we worked in this case with smaller patches (128×128) with 80% overlap, which contributes to increase the inference time.

Table 5.7: Average processing time for each FCN for the Deforestation Dataset.

Method	**Training Time (min:s)**	**Inference Time (s)**
U-Net	07:47	29.9
SegNet	22:17	34.4
FC-DenseNet	9:49	38.6
DeepLabv3+ (Xception)	9:28	36.3
DeepLabv3+ (MobileNetV2)	10:13	25.3

6
CONCLUSIONS

In this work, we evaluated five state-of-the-art fully convolutional networks for semantic segmentation on different datasets composed of RGB and multispectral images acquired by UAV, airplane, and satellite platforms. Five architectures were tested: SegNet, U-Net, FC-DenseNet, and two DeepLabv3+ variants, specifically Xception and MobileNetV2. The analysis was conducted on two datasets that represent an urban context and one represented a forested area. The experiments demonstrated that networks could learn the distinguishing features of the target class in a supervised way. This fact indicated that the tested FCN designs could delineate several targets, provided that enough representative labeled samples are available for training.

In the first set of experiments, on the cumbaru dataset, FC-DenseNet attained the best performance among the tested networks. Ranked second and third were U-Net and DeepLabv3+ MobileNetV2, respectively, with a difference of 1.4%, 1.7%, and 3.1% in terms of Overall Accuracy, F1-score, and IoU, followed by SegNet. The lowest accuracy scores were achieved by DeepLabv3+ Xception.

We also observed in our study that post-processing the networks' outcomes with a fully connected CRF was beneficial in this application in nearly all cases. However, the numerical impact on the overall accuracy metrics was often modest, because CRF generally fixes errors in small image regions. Yet, the improvement in segmentation quality was usually significant, as evidenced by visual inspection. The price for such accuracy gain was the comparatively long CRF processing time, about 30 times the FC-DenseNet's inference time. We also noticed that DeepLabv3+ Xception benefited from CRF more than all other architectures.

For the segmentation of Rooftops using aircraft images, DeepLabv3+ variants were the best performing network in terms of overall accuracy, F1-score, and IoU, respectively. FC-DenseNet and SegNet ranked in third and fourth places, respectively, with similar but slightly difference values between the metrics, above 1.5%. U-Net exhibits the lowest quantitative values for all of the accuracy metrics.

After, the post-processing step we noticed in our experiments that

fully connected CRF reduces false positive and false negative samples in the prediction results of all of the FCNs. However, it shows a negative effect on the accuracy metrics for all the models. Typical downsides is that CRF tend to eliminate small labeled regions and merges some roofs that are very close to each other. Moreover, similar to UAV dataset, the CRF post-processing stage presumes a longer processing time.

In the last application we proposed the use of the aforementioned FCN models to handle spatial and temporal information of satellite images to detect yearly changes in the vegetation cover of the Brazilian Amazon. The experiments were conducted on two Landsat-8 images acquired one year apart from each other. The work shows a great potential of the methods in identify deforestation samples with a small training data with augmentation in the form of overlapping sample patches; keeping a faster training with a high accuracy in the prediction maps. In this case, U-Net and FC-DenseNet presented the best performance in most experiments. The FCNs also produced predictions masks with well-defined deforestation samples. Among the models, U-Net and FC-DenseNet shows a lower tendency to produce false-negative samples.

We also include in the study the metric Alarm Area as a function to evaluate the human interaction in the visual interpretation task of deforested areas. Here, the analysis is restricted to a small portion of the image as the probability threshold varies. Results show that 90% of deforestation examples are present in only 10% of the total imaged area.

In general, all tested networks showed good performance for the segmentation of diverse targets, different platforms, and a varied proportion of the target class. Although, in our experiments, we cannot establish a superiority of one network over another as the behaviour of each FCN differs as the scenario changes. For instance, FC-DenseNet and U-Net presented the best performance in the UAV and Satellite datasets, respectively. Although, for the Rooftop Segmentation, the DeepLabv3+ with the MobileNetv2 encoder consistently achieve better performance on the test set among all the FCN models.

Based on the comparison of different remote sensing platforms, it is worth to mention that the very high resolution of UAV provide higher accuracy results among the different platforms.

In the continuation of this research, we intend to verify if CRF is generally able to mitigate the problem of scarce training data for FCN based semantic segmentation. Additionally, we aim to investigate the application of morphological algorithm as an alternative to CRF. Another issue that deserves further analysis concerns the generalization of these network designs.

Unfortunately, the number and diversity of annotated databases available for this purpose are still limited. We are currently working on building a more diverse database in terms of sensors and climate characteristics.

Bibliography

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

2 KRIZHEVSKY, A.; SUTSKEVER, I. ; HINTON, G. E.. **Imagenet classification with deep convolutional neural networks**. In: PROCEEDINGS OF THE 25TH INTERNATIONAL CONFERENCE ON NEURAL INFORMATION PROCESSING SYSTEMS - VOLUME 1, NIPS'12, p. 1097–1105, Red Hook, NY, USA, 2012. Curran Associates Inc.

3 IRONS, J. R.; DWYER, J. L. ; BARSI, J. A.. **The next landsat satellite: The landsat data continuity mission**. Remote Sensing of Environment, 122:11 – 21, 2012. Landsat Legacy Special Issue.

4 BERMUDEZ, J. D.; HAPP, P. N.; FEITOSA, R. Q. ; OLIVEIRA, D. A. B.. **Synthesis of multispectral optical images from sar/optical multitemporal data using conditional generative adversarial networks**. IEEE Geoscience and Remote Sensing Letters, 16(8):1220–1224, 2019.

5 TOTH, C.; JÓŹKÓW, G.. **Remote sensing platforms and sensors: A survey**. ISPRS Journal of Photogrammetry and Remote Sensing, 115:22–36, 2016.

6 MATESE, A.; TOSCANO, P.; DI GENNARO, S. F.; GENESIO, L.; VACCARI, F. P.; PRIMICERIO, J.; BELLI, C.; ZALDEI, A.; BIANCONI, R. ; GIOLI, B.. **Intercomparison of uav, aircraft and satellite remote sensing platforms for precision viticulture**. Remote Sensing, 7(3):2971–2990, 2015.

7 CHENG, G.; HAN, J.. **A survey on object detection in optical remote sensing images**. ISPRS Journal of Photogrammetry and Remote Sensing, 117:11 – 28, 2016.

8 FASSNACHT, F.; LATIFI, H.; STEREŃCZAK, K.; MODZELEWSKA, A.; LEFSKY, M.; WASER, L.; STRAUB, C. ; GHOSH, A.. **Review of studies**

on tree species classification from remotely sensed data. Remote Sensing of Environment, 186:64–87, 08 2016.

9 SALOVAARA, K.; THESSLER, S.; MALIK, R. ; TUOMISTO, H.. **Classification of amazonian primary rain forest vegetation using landsat etm+satellite imagery**. Remote Sensing of Environment, 97:39–51, 07 2005.

10 ROGAN, J.; MILLER, J.; STOW, D.; FRANKLIN, J.; LEVIEN, L. ; FISCHER, C.. **Land-cover change monitoring with classification trees using landsat tm and ancillary data**. Photogrammetric Engineering and Remote Sensing, 69, 07 2003.

11 SANTOS, A. A. D.; MARCATO JUNIOR, J.; ARAÚJO, M. S.; DI MARTINI, D. R.; TETILA, E. C.; SIQUEIRA, H. L.; AOKI, C.; ELTNER, A.; MATSUBARA, E. T.; PISTORI, H.; FEITOSA, R. Q.; LIESENBERG, V. ; GONÇALVES, W. N.. **Assessment of cnn-based methods for individual tree detection on images captured by rgb cameras attached to uavs**. Sensors, 19(16), 2019.

12 MOTTAGHI, R.; CHEN, X.; LIU, X.; CHO, N.-G.; LEE, S.-W.; FIDLER, S.; URTASUN, R. ; YUILLE, A.. **The role of context for object detection and semantic segmentation in the wild**. In: PROCEEDINGS OF THE IEEE CONFERENCE ON COMPUTER VISION AND PATTERN RECOGNITION, p. 891–898, 2014.

13 CHEN, X.; MOTTAGHI, R.; LIU, X.; FIDLER, S.; URTASUN, R. ; YUILLE, A. L.. **Detect what you can: Detecting and representing objects using holistic models and body parts**. CoRR, abs/1406.2031, 2014.

14 BADRINARAYANAN, V.; HANDA, A. ; CIPOLLA, R.. **Segnet: A deep convolutional encoder-decoder architecture for robust semantic pixel-wise labelling**. CoRR, abs/1505.07293, 2015.

15 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.

16 LI, Y.; ZHANG, H.; XUE, X.; JIANG, Y. ; SHEN, Q.. **Deep learning for remote sensing image classification: A survey**. Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery, p. e1264, 05 2018.

17 YIN, Z.; MA, A. ; GOLDBERG, D. W.. **A deep learning approach for rooftop geocoding**. Transactions in GIS, 23(3):495–514, 2019.

18 CASTAGNO, J.; ATKINS, E.. **Roof shape classification from lidar and satellite image data fusion using supervised learning**. Sensors, 18(11):3960, 2018.

19 LONG, J.; SHELHAMER, E. ; DARRELL, T.. **Fully convolutional networks for semantic segmentation**. CoRR, abs/1411.4038, 2014.

20 RONNEBERGER, O.; FISCHER, P. ; BROX, T.. **U-net: Convolutional networks for biomedical image segmentation**. CoRR, abs/1505.04597, 2015.

21 LIU, Y.; PIRAMANAYAGAM, S.; MONTEIRO, S. T. ; SABER, E.. **Dense semantic labeling of very-high-resolution aerial imagery and lidar with fully-convolutional neural networks and higher-order crfs**. IEEE Conference on Computer Vision and Pattern Recognition (CVPR), p. 1561–1570, 07 2017.

22 LIU, Y.; FAN, B.; WANG, L.; BAI, J.; XIANG, S. ; PAN, C.. **Semantic labeling in very high resolution images via a self-cascaded convolutional neural network**. ISPRS journal of photogrammetry and remote sensing, 145:78–95, 2018.

23 CHEN, L.; PAPANDREOU, G.; SCHROFF, F. ; ADAM, H.. **Rethinking atrous convolution for semantic image segmentation**. CoRR, abs/1706.05587, 2017.

24 ARAKAKI, A. A. H.; SCHEIDT, G. N.; PORTELLA, A. C.; ARRUDA, E. J. A. D. ; COSTA, R. B. D.. **O baru (dipteryx alata vog.) como alternativa de sustentabilidade em área de fragmento florestal do cerrado, no mato grosso do sul**. Interaçõ̃es (Campo Grande), 10:31 – 39, 06 2009.

25 ROGAN, J.; MILLER, J.; WULDER, M. ; FRANKLIN, S.. **Integrating gis and remotely sensed data for mapping forest disturbance and change**. Understanding forest disturbance and spatial pattern: Remote sensing and GIS approaches, p. 133–172, 2006.

26 MANCINO, G.; NOLÈ, A.; RIPULLONE, F. ; FERRARA, A.. **Landsat tm imagery and ndvi differencing to detect vegetation change: assessing natural forest expansion in basilicata, southern italy**. iForest-Biogeosciences and Forestry, 7(2):75, 2014.

27 RAWAT, J.; KUMAR, M.. **Monitoring land use/cover change using remote sensing and gis techniques: A case study of hawalbagh block, district almora, uttarakhand, india**. The Egyptian Journal of Remote Sensing and Space Science, 18(1):77–84, 2015.

28 CHAND, T. K.; BADARINATH, K.; PRASAD, V. K.; MURTHY, M.; ELVIDGE, C. D. ; TUTTLE, B. T.. **Monitoring forest fires over the indian region using defense meteorological satellite program-operational linescan system nighttime satellite data**. Remote Sensing of Environment, 103(2):165–178, 2006.

29 MUCHONEY, D. M.; HAACK, B. N.. **Change detection for monitoring forest defoliation**. Photogrammetric engineering and remote sensing, 60(10):1243–1252, 1994.

30 SUNAR, F.. **An analysis of changes in a multi-date data set: a case study in the ikitelli area, istanbul, turkey**. International Journal of Remote Sensing, 19(2):225–235, 1998.

31 PRAKASH, A.; GUPTA, R.. **Land-use mapping and change detection in a coal mining area-a case study in the jharia coalfield, india**. International journal of remote sensing, 19(3):391–410, 1998.

32 ASNER, G. P.; KELLER, M.; PEREIRA JR, R. ; ZWEEDE, J. C.. **Remote sensing of selective logging in amazonia: Assessing limitations based on detailed field observations, landsat etm+, and textural analysis**. Remote Sensing of Environment, 80(3):483–496, 2002.

33 SINGH, A.. **Change detection in the tropical forest environment of northeastern india using landsat**. Remote sensing and tropical land management, p. 237–254, 1986.

34 LU, D.; MAUSEL, P.; BRONDIZIO, E. ; MORAN, E.. **Change detection techniques**. International journal of remote sensing, 25(12):2365–2401, 2004.

35 NELSON, R. F.. **Detecting forest canopy change due to insect activity using landsat mss**. Photogrammetric Engineering and Remote Sensing, 49(9):1303–1314, 1983.

36 HAYES, D. J.; SADER, S. A.. **Comparison of change-detection techniques for monitoring tropical forest clearing and vegetation regrowth in a time series**. Photogrammetric engineering and remote sensing, 67(9):1067–1075, 2001.

37 ZHU, Z.; WOODCOCK, C. E.. **Continuous change detection and classification of land cover using all available landsat data**. Remote Sensing of Environment, 144:152 – 171, 2014.

38 IM, J.; JENSEN, J. R.. **A change detection model based on neighborhood correlation image analysis and decision tree classification**. Remote Sensing of Environment, 99(3):326 – 340, 2005.

39 SCHNEIDER, A.. **Monitoring land cover change in urban and peri-urban areas using dense time stacks of landsat satellite data and a data mining approach**. Remote Sensing of Environment, 124:689–704, 2012.

40 HUANG, C.; SONG, K.; KIM, S.; TOWNSHEND, J. R.; DAVIS, P.; MASEK, J. G. ; GOWARD, S. N.. **Use of a dark object concept and support vector machines to automate forest cover change analysis**. Remote sensing of environment, 112(3):970–985, 2008.

41 TOWNSHEND, J.; HUANG, C.; KALLURI, S.; DEFRIES, R.; LIANG, S. ; YANG, K.. **Beware of per-pixel characterization of land cover**. International Journal of remote sensing, 21(4):839–843, 2000.

42 KHAN, S. H.; HE, X.; PORIKLI, F. ; BENNAMOUN, M.. **Forest change detection in incomplete satellite images with deep neural networks**. IEEE Transactions on Geoscience and Remote Sensing, 55(9):5407–5423, 2017.

43 MOU, L.; BRUZZONE, L. ; ZHU, X. X.. **Learning spectral-spatial-temporal features via a recurrent convolutional neural network for change detection in multispectral imagery**. IEEE Transactions on Geoscience and Remote Sensing, 57(2):924–935, 2019.

44 LIU, Y.; CHEN, X.; WANG, Z.; WANG, Z. J.; WARD, R. K. ; WANG, X.. **Deep learning for pixel-level image fusion: Recent advances and future prospects**. Information Fusion, 42:158–173, 2018.

45 LIU, X.; LIU, Q. ; WANG, Y.. **Remote sensing image fusion based on two-stream fusion network**. Information Fusion, 55:1–15, 2020.

46 SCARPA, G.; VITALE, S. ; COZZOLINO, D.. **Target-adaptive cnn-based pansharpening**. IEEE Transactions on Geoscience and Remote Sensing, 56(9):5443–5457, 2018.

47 HOU, B.; WANG, Y. ; LIU, Q.. **Change detection based on deep features and low rank**. IEEE Geoscience and Remote Sensing Letters, 14(12):2418–2422, 2017.

48 DAUDT, R. C.; LE SAUX, B.; BOULCH, A. ; GOUSSEAU, Y.. **Urban change detection for multispectral earth observation using convolutional neural networks**. In: IGARSS 2018-2018 IEEE INTERNATIONAL GEOSCIENCE AND REMOTE SENSING SYMPOSIUM, p. 2115–2118. IEEE, 2018.

49 ORTEGA, M.; BERMUDEZ, J.; HAPP, P.; GOMES, A. ; FEITOSA, R.. **Evaluation of deep learning techniques for deforestation detection in the amazon forest.** ISPRS Annals of Photogrammetry, Remote Sensing & Spatial Information Sciences, 4, 2019.

50 ZHANG, W.; LU, X.. **The spectral-spatial joint learning for change detection in multispectral imagery**. Remote Sensing, 11(3):240, 2019.

51 CAYE DAUDT, R.; LE SAUX, B. ; BOULCH, A.. **Fully convolutional siamese networks for change detection**. In: 2018 25TH IEEE INTERNATIONAL CONFERENCE ON IMAGE PROCESSING (ICIP), p. 4063–4067, 2018.

52 LEI, T.; ZHANG, Y.; LV, Z.; LI, S.; LIU, S. ; NANDI, A. K.. **Landslide inventory mapping from bitemporal images using deep convolutional neural networks**. IEEE Geoscience and Remote Sensing Letters, 16(6):982–986, 2019.

53 DE BEM, P. P.; DE CARVALHO JUNIOR, O. A.; FONTES GUIMARÃES, R. ; TRANCOSO GOMES, R. A.. **Change detection of deforestation in the brazilian amazon using landsat data and convolutional neural networks**. Remote Sensing, 12(6):901, 2020.

54 PENG, D.; ZHANG, Y. ; GUAN, H.. **End-to-end change detection for high resolution satellite images using improved unet++**. Remote Sensing, 11(11):1382, 2019.

55 BISCHOF, H.; SCHNEIDER, W. ; PINZ, A. J.. **Multispectral classification of landsat-images using neural networks**. IEEE transactions on Geoscience and Remote Sensing, 30(3):482–490, 1992.

56 ABRAHAM, L.; SASIKUMAR, M.. **Automatic building extraction from satellite images using artificial neural networks**. Procedia Engineering, 50:893–903, 2012.

57 JOSHI, B.; BALUYAN, H.; HINAI, A. A. ; WOON, W. L.. **Automatic rooftop detection using a two-stage classification**. In: 2014 UKSIM-AMSS 16TH INTERNATIONAL CONFERENCE ON COMPUTER MODELLING AND SIMULATION, p. 286–291. IEEE, 2014.

58 GIRSHICK, R.; DONAHUE, J.; DARRELL, T. ; MALIK, J.. **Rich feature hierarchies for accurate object detection and semantic segmentation**. In: THE IEEE CONFERENCE ON COMPUTER VISION AND PATTERN RECOGNITION (CVPR), June 2014.

59 VAKALOPOULOU, M.; KARANTZALOS, K.; KOMODAKIS, N. ; PARAGIOS, N.. **Building detection in very high resolution multispectral data with deep learning features**. In: 2015 IEEE INTERNATIONAL GEOSCIENCE AND REMOTE SENSING SYMPOSIUM (IGARSS), p. 1873–1876. IEEE, 2015.

60 SOLOVYEV, R.. **Roof material classification from aerial imagery**, 2020.

61 SOMAN, K.. **Rooftop detection using aerial drone imagery**. In: PROCEEDINGS OF THE ACM INDIA JOINT INTERNATIONAL CONFERENCE ON DATA SCIENCE AND MANAGEMENT OF DATA, p. 281–284, 2019.

62 ZHONG, Z.; LI, J.; CUI, W. ; JIANG, H.. **Fully convolutional networks for building and road extraction: Preliminary results**. In: 2016 IEEE INTERNATIONAL GEOSCIENCE AND REMOTE SENSING SYMPOSIUM (IGARSS), p. 1591–1594, 2016.

63 XU, Y.; WU, L.; XIE, Z. ; CHEN, Z.. **Building extraction in very high resolution remote sensing imagery using deep learning and guided filters**. Remote Sensing, 10:144, 01 2018.

64 WU, G.; SHAO, X.; GUO, Z.; CHEN, Q.; YUAN, W.; SHI, X.; XU, Y. ; SHIBASAKI, R.. **Automatic building segmentation of aerial imagery using multi-constraint fully convolutional networks**. Remote Sensing, 10:407, 03 2018.

65 MOU, L.; HUA, Y. ; ZHU, X. X.. **A relation-augmented fully convolutional network for semantic segmentation in aerial scenes**. CoRR, abs/1904.05730, 2019.

66 MNIH, V.. **Machine learning for aerial image labeling**. Citeseer, 2013.

67 KLUCKNER, S.; BISCHOF, H.. **Image-based building classification and 3d modeling with super-pixels**. International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences - ISPRS Archives, 38, 06 2010.

68 GERKE, M.. **Use of the stair vision library within the isprs 2d semantic labeling benchmark (vaihingen)**. 01 2015.

69 PAISITKRIANGKRAI, S.; SHERRAH, J.; JANNEY, P.; HENGEL, V.-D. ; OTHERS. **Effective semantic pixel labelling with convolutional networks and conditional random fields**. In: PROCEEDINGS OF THE IEEE CONFERENCE ON COMPUTER VISION AND PATTERN RECOGNITION WORKSHOPS, p. 36–43, 2015.

70 ALONZO, M.; BOOKHAGEN, B. ; ROBERTS, D.. **Urban tree species mapping using hyperspectral and lidar data fusion**. Remote Sensing of Environment, 148:70–83, 05 2014.

71 BROCKHAUS, J. A.; KHORRAM, S.. **A comparison of spot and landsat-tm data for use in conducting inventories of forest resources**. International Journal of Remote Sensing, 13(16):3035–3043, 1992.

72 JOHANSEN, K.; PHINN, S.. **Mapping structural parameters and species composition of riparian vegetation using ikonos and rogan2006integratinglandsat etm+ data in australian tropical savannahs**. Photogrammetric Engineering and Remote Sensing, 72, 01 2006.

73 CLARK, M.; ROBERTS, D. ; CLARK, D.. **Hyperspectral discrimination of tropical rain forest tree species at leaf to crown scales**. Remote Sensing of Environment, 96:375–398, 06 2005.

74 ZHEN, Z.; QUACKENBUSH, L. J. ; ZHANG, L.. **Trends in automatic individual tree crown detection and delineation—evolution of lidar data**. Remote Sensing, 8(4), 2016.

75 WANG, K.; WANG, T. ; LIU, X.. **A review: Individual tree species classification using integrated airborne lidar and optical imagery with a focus on the urban environment**. Forests, 10(1):1–18, 12 2018.

76 GONG, P.; HOWARTH, P.. **An assessment of some factors influencing multispectral land-cover classification.** Photogramm. Eng. Remote Sens., 56:597–603, 1990.

77 CHENARI, A.; ERFANIFARD, Y.; DEHGHANI, M. ; POURGHASEMI, H. R.. **Woodland mapping at single-tree levels using object-oriented classification of unmanned aerial vehicle (uav) images**. ISPRS - International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences, XLII-4/W4:43–49, 09 2017.

78 FENG, X.; LI, P.. **A tree species mapping method from uav images over urban area using similarity in tree-crown object histograms**. Remote Sensing, 11:1982, 08 2019.

79 BAENA, S.; MOAT, J.; WHALEY, O. ; BOYD, D. S.. **Identifying species from the air: Uavs and the very high resolution challenge for plant conservation**. PloS one, 12(11):e0188714, 2017.

80 LI, W.; FU, H.; YU, L. ; CRACKNELL, A.. **Deep learning based oil palm tree detection and counting for high-resolution remote sensing images**. Remote Sensing, 9(1), 2017.

81 WEINSTEIN, B. G.; MARCONI, S.; BOHLMAN, S.; ZARE, A. ; WHITE, E.. **Individual tree-crown detection in rgb imagery using semi-supervised deep learning neural networks**. Remote Sensing, 11(11), 2019.

82 NATESAN, S.; ARMENAKIS, C. ; VEPAKOMMA, U.. **Resnet-based tree species classification using uav images.** International Archives of the Photogrammetry, Remote Sensing & Spatial Information Sciences, 2019.

83 ONISHI, M.; ISE, T.. **Automatic classification of trees using a UAV onboard camera and deep learning**. CoRR, abs/1804.10390, 2018.

84 BAATZ, M.; SCHAPE, A.. **Multiresolution segmentation-an optimization approach for high quality multi scale image segmentation**. Angewandte Geographische Informationsverarbeitung, XII:12–23, 2000.

85 WAGNER, F. H.; SANCHEZ, A.; TARABALKA, Y.; LOTTE, R. G.; FERREIRA, M. P.; AIDAR, M. P.; GLOOR, E.; PHILLIPS, O. L. ; ARAGAO, L. E.. **Using the u-net convolutional network to map forest types and disturbance in the atlantic rainforest with very high resolution images**. Remote Sensing in Ecology and Conservation, 2019.

86 KATTENBORN, T.; EICHEL, J. ; FASSNACHT, F. E.. **Convolutional neural networks enable efficient, accurate and fine-grained**

segmentation of plant species and communities from high-resolution uav imagery. Scientific reports, 9(1):1–9, 2019.

87 HILLEL, D.; HATFIELD, J. L.. **Encyclopedia of soils in the environment**, volumen 3. Elsevier Amsterdam, 2005.

88 MUJTABA, G.; KARAM, F. W.. **Monitoring deforestation using remote sensing**. International Journal of Computer Science and Information Security, 15(1):75, 2017.

89 BETTINGER, P.; BOSTON, K.; SIRY, J. P. ; GREBNER, D. L.. **Chapter 3 - geographic information and land classification in support of forest planning**. In: Bettinger, P.; Boston, K.; Siry, J. P. ; Grebner, D. L., editors, FOREST MANAGEMENT AND PLANNING (SECOND EDITION), p. 65 – 85. Academic Press, second edition edition, 2017.

90 FITZGERALD, G. J.. **Characterizing vegetation indices derived from active and passive sensors**. International Journal of Remote Sensing, 31(16):4335–4348, 2010.

91 XIANG, T.; XIA, G. ; ZHANG, L.. **Mini-uav-based remote sensing: Techniques, applications and prospectives**. CoRR, abs/1812.07770, 2018.

92 MATESE, A.; TOSCANO, P.; DI GENNARO, S. F.; GENESIO, L.; VACCARI, F. P.; PRIMICERIO, J.; BELLI, C.; ZALDEI, A.; BIANCONI, R. ; GIOLI, B.. **Intercomparison of uav, aircraft and satellite remote sensing platforms for precision viticulture**. Remote Sensing, 7(3):2971–2990, 2015.

93 ALBAWI, S.; MOHAMMED, T. A. ; AL-ZAWI, S.. **Understanding of a convolutional neural network**. In: 2017 INTERNATIONAL CONFERENCE ON ENGINEERING AND TECHNOLOGY (ICET), p. 1–6. IEEE, 2017.

94 IOFFE, S.; SZEGEDY, C.. **Batch normalization: Accelerating deep network training by reducing internal covariate shift**. CoRR, abs/1502.03167, 2015.

95 LIU, Y.; NGUYEN, D.; DELIGIANNIS, N.; DING, W. ; MUNTEANU, A.. **Hourglass-shapenetwork based semantic segmentation for high resolution aerial imagery**. Remote Sensing, 9:522, 05 2017.

96 VOLPI, M.; TUIA, D.. **Dense semantic labeling of sub-decimeter resolution images with convolutional neural networks**. CoRR, abs/1608.00775, 2016.

97 CHEN, L.-C.; PAPANDREOU, G.; KOKKINOS, I.; MURPHY, K. ; YUILLE, A. L.. **Semantic image segmentation with deep convolutional nets and fully connected crfs**. CoRR, abs/1412.7062, 2014.

98 CHEN, L.; ZHU, Y.; PAPANDREOU, G.; SCHROFF, F. ; ADAM, H.. **Encoder-decoder with atrous separable convolution for semantic image segmentation**. CoRR, abs/1802.02611, 2018.

99 LI, Y.; ZHANG, X. ; CHEN, D.. **Csrnet: Dilated convolutional neural networks for understanding the highly congested scenes**. CoRR, abs/1802.10062, 2018.

100 CHOLLET, F.. **Xception: Deep learning with depthwise separable convolutions**. CoRR, abs/1610.02357, 2016.

101 GARCIA-GARCIA, A.; ORTS-ESCOLANO, S.; OPREA, S.; VILLENA-MARTINEZ, V. ; RODRÍGUEZ, J. G.. **A review on deep learning techniques applied to semantic segmentation**. CoRR, abs/1704.06857, 2017.

102 JÉGOU, S.; DROZDZAL, M.; VÁZQUEZ, D.; ROMERO, A. ; BENGIO, Y.. **The one hundred layers tiramisu: Fully convolutional densenets for semantic segmentation**. 2017 IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW), p. 1175–1183, 2016.

103 HUANG, G.; LIU, Z. ; WEINBERGER, K. Q.. **Densely connected convolutional networks**. CoRR, abs/1608.06993, 2016.

104 HAI, J.; QIAO, K.; CHEN, J. J.; TAN, H.; XU, J.; ZENG, L.; SHI, D. ; YAN, B.. **Fully convolutional densenet with multiscale context for automated breast tumor segmentation**. In: JOURNAL OF HEALTHCARE ENGINEERING, p. 11, 2019.

105 HARIHARAN, B.; ARBELÁEZ, P. A.; GIRSHICK, R. B. ; MALIK, J.. **Hypercolumns for object segmentation and fine-grained localization**. CoRR, abs/1411.5752, 2014.

106 ZHANG, T.; JIANG, S.; ZHAO, Z.; DIXIT, K.; ZHOU, X.; HOU, J.; ZHANG, Y. ; YAN, C.. **Rapid and robust two-dimensional phase**

unwrapping via deep learning. Opt. Express, 27(16):23173–23185, Aug 2019.

107 SANDLER, M.; HOWARD, A. G.; ZHU, M.; ZHMOGINOV, A. ; CHEN, L.. **Inverted residuals and linear bottlenecks: Mobile networks for classification, detection and segmentation**. CoRR, abs/1801.04381, 2018.

108 LIU, Y.; REN, Q.; GENG, J.; DING, M. ; LI, J.. **Efficient patch-wise semantic segmentation for large-scale remote sensing images**. In: SENSORS, volumen 18, p. 3232, 2018.

109 CHEN, L.-C.; PAPANDREOU, G.; KOKKINOS, I.; MURPHY, K. ; YUILLE, A. L.. **Semantic image segmentation with deep convolutional nets and fully connected crfs**. CoRR, abs/1412.7062, 2014.

110 KRÄHENBÜHL, P.; KOLTUN, V.. **Efficient inference in fully connected crfs with gaussian edge potentials**. In: PROCEEDINGS OF THE 24TH INTERNATIONAL CONFERENCE ON NEURAL INFORMATION PROCESSING SYSTEMS, NIPS'11, p. 109–117, USA, 2011. Curran Associates Inc.

111 COCHRANE, T.; COCHRANE, T. A.. **Diversity of the land resources in the Amazonian State of RondÃ´nia, Brazil**. Acta Amazonica, 36:91 – 101, 03 2006.

112 SOKOLOVA, M.; JAPKOWICZ, N. ; SZPAKOWICZ, S.. **Beyond accuracy, f-score and roc: a family of discriminant measures for performance evaluation**. In: AUSTRALASIAN JOINT CONFERENCE ON ARTIFICIAL INTELLIGENCE, p. 1015–1021. Springer, 2006.

113 REZATOFIGHI, S. H.; TSOI, N.; GWAK, J.; SADEGHIAN, A.; REID, I. D. ; SAVARESE, S.. **Generalized intersection over union: A metric and A loss for bounding box regression**. CoRR, abs/1902.09630, 2019.

114 CHOLLET, F.; OTHERS. **Keras**. `https://keras.io`, 2015.

115 KINGMA, D. P.; BA, J.. **Adam: A method for stochastic optimization**. CoRR, abs/1412.6980, 2014.