remote sensing

MDPI

Article

Instance Segmentation for Large, Multi-Channel Remote Sensing Imagery Using Mask-RCNN and a Mosaicking Approach

Osmar Luiz Ferreira de Carvalho [1], Osmar Abílio de Carvalho Júnior [2,*], Anesmar Olino de Albuquerque [2], Pablo Pozzobon de Bem [2], Cristiano Rosa Silva [1], Pedro Henrique Guimarães Ferreira [1], Rebeca dos Santos de Moura [2], Roberto Arnaldo Trancoso Gomes [2], Renato Fontes Guimarães [2] and Díbio Leandro Borges [3]

[1] Departamento de Engenharia Elétrica, Campus Universitário Darcy Ribeiro, Asa Norte, Universidade de Brasília, DF, Brasília 70910-900, Brazil; osmarcarvalho@ieee.org (O.L.F.d.C.); cristiano@dubbox.com.br (C.R.S.); pedroferreira@ieee.org (P.H.G.F.)

[2] Departamento de Geografia, Campus Universitário Darcy Ribeiro, Asa Norte, Universidade de Brasília, DF, Brasília 70910-900, Brazil; anesmar@ieee.org (A.O.d.A.); pablo.bem@aluno.unb.br (P.P.d.B.); moura.santos@aluno.unb.br (R.d.S.d.M.); robertogomes@unb.br (R.A.T.G.); renatofg@unb.br (R.F.G.)

[3] Departamento de Ciência da Computação, Campus Universitário Darcy Ribeiro, Asa Norte, Universidade de Brasília, DF, Brasília 70910-900, Brazil; dibio@unb.br

* Correspondence: osmarjr@unb.br

check for updates

Citation: Carvalho, O.L.F.d.; de Carvalho Júnior, O.A.; Albuquerque, A.O.d.; Bem, P.P.d.; Silva, C.R.; Ferreira, P.H.G.; Moura, R.d.S.d.; Gomes, R.A.T.; Guimarães, R.F.; Borges, D.L. Instance Segmentation for Large, Multi-Channel Remote Sensing Imagery Using Mask-RCNN and a Mosaicking Approach. *Remote Sens.* **2021**, *13*, 39. https://dx.doi.org/10.3390/rs13010039

Received: 1 December 2020
Accepted: 20 December 2020
Published: 24 December 2020

Abstract: Instance segmentation is the state-of-the-art in object detection, and there are numerous applications in remote sensing data where these algorithms can produce significant results. Nevertheless, one of the main problems is that most algorithms use Red, Green, and Blue (RGB) images, whereas Satellite images often present more channels that can be crucial to improve performance. Therefore, the present work brings three contributions: (a) conversion system from ground truth polygon data into the Creating Common Object in Context (COCO) annotation format; (b) Detectron2 software source code adaptation and application on multi-channel imagery; and (c) large scene image mosaicking. We applied the procedure in a Center Pivot Irrigation System (CPIS) dataset with ground truth produced by the Brazilian National Water Agency (ANA) and Landsat-8 Operational Land Imager (OLI) imagery (7 channels with 30-m resolution). Center pivots are a modern irrigation system technique with massive growth potential in Brazil and other world areas. The round shapes with different textures, colors, and spectral behaviors make it appropriate to use Deep Learning instance segmentation. We trained the model using 512 × 512-pixel sized patches using seven different backbone structures (ResNet50- Feature Pyramid Network (FPN), Resnet50-DC5, ResNet50-C4, Resnet101-FPN, Resnet101-DC5, ResNet101-FPN, and ResNeXt101-FPN). The model evaluation used standard COCO metrics (Average Precision (AP), AP_{50}, AP_{75}, AP_{small}, AP_{medium}, and AR_{100}). ResNeXt101-FPN had the best results, with a 3% advantage over the second-best model (ResNet101-FPN). We also compared the ResNeXt101-FPN model in the seven-channel and RGB imagery, where the multi-channel model had a 3% advantage, demonstrating great improvement using a larger number of channels. This research is also the first with a mosaicking algorithm using instance segmentation models, where we tested in a 1536 × 1536-pixel image using a non-max suppression sorted by area method. The proposed methodology is innovative and suitable for many other remote sensing problems and medical imagery that often present more channels.

Keywords: instance segmentation; multi-channel imagery; mask R-CNN; deep learning; COCO; Landsat-8; center pivot

1. Introduction

In the last few years, Deep Learning (DL) became the most used method in computer vision and object detection problems in remote sensing imagery [1–3]. DL enables pattern recognition in different data abstraction levels, varying from low-level information (corners and edges), up to high-level information (full objects) [4]. This approach achieves state-of-the-art results in different applications in remote sensing digital image processing [5]: pan-sharpening [6–9]; image registration [10–13], change detection [14–17], object detection [18–21], semantic segmentation [22–25], and time series analysis [26–29]. The classification algorithms applied in remote sensing imagery uses spatial, spectral, and temporal information to extract characteristics from the targets, where a wide variety of targets show significant results: clouds [30–33], dust-related air pollutant [34–37] land-cover/land-use [38–41], urban features [42–45], and ocean [46–49], among others.

The DL techniques regarding segmentation have two subdivisions: (i) semantic segmentation (labels are class-aware); and (ii) instance segmentation (labels are instance-aware). Semantic segmentation brings pixel-wise classification to the entire scene, with pieces of information about the category, localization, and shape [50]. In addition, semantic segmentation differs from image classification since it enables all object parts to interact, by identifying and grouping pixels that are semantically together [51]. The deep semantic understanding allows us to aggregate the different parts in the formation of a whole, considering variations of colors, textures, and patterns. Several reviews on semantic segmentation published recently, highlights the algorithms' innovations, applications, and taxonomy [51–55].

However, the semantic segmentation results do not distinguish different instances within the same category, resulting in limitations in individually separating objects. Therefore, this new problem is not only to determine the pixels of a specific class (semantic segmentation) but also includes the discernment of different objects in the same category by obtaining the exact number of a given object in the image (instance segmentation). Therefore, instance segmentation consists of a new paradigm and evolution of semantic segmentation by allowing a unique understanding of each object, counting the number of objects, and analyzing objects in occlusion and contact conditions.

Instance segmentation algorithms have two main approaches [56]: (a) segmentation-first strategy, where segmentations occur before classification, and (b) instance-first strategy, parallel process of both segmentation and classification. In turn, the segmentation-first strategy also has two approaches: (a) segment-based, first establishes segment candidates and then performs their classification [57–59]; and (b) based on semantic segmentation masks, trying to separate the pixels of the same classes in different instances [60–63].

The instance-first strategy methods have advantages for being more straightforward and more flexible, allowing the algorithm to obtain the bounding boxes and the segmentation masks simultaneously. The main models proposed were Fully Convolutional Instance-Aware Semantic Segmentation (FCIS) [64], Mask-Region-based Convolutional Neural Network (Mask R-CNN) [56], Cascade Mask R-CNN [65,66], Mask Scoring R-CNN [67], and High-Quality Instance Segmentation Network (HQ-ISNet) (based on Cascade Mask R-CNN) [68].

Instance segmentation has applications in several areas of knowledge: medicine [69,70], biology [71,72], livestock [73,74], agronomy [75,76], among others. However, remote sensing application is still restricted, highlighting its use in the automatic detection of the following targets: marine oil spill [77], building [78,79], vehicle [80], and ship [81].

However, surpassing some challenges is necessary for a broad application of instance segmentation in remote sensing (and multi-channel medical imagery). The instance segmentation frameworks (e.g., Detectron2) use configurations and libraries with restricted compatibility with Red, Green, and Blue (RGB) images, traditionally applied by the computer vision community in tasks, such as fruit detection [82] and animal recognition [83], among others. This is a data limitation for optical Earth observation sensors that are generally multispectral, where the available channels provide complementary information that maximizes accuracy. In semantic segmentation, approaches to aggregate more information considered: (a) the use of image fusion techniques, where the three bands used are data integration products [84]; (b) input layer adequation to support a larger amount of channels, e.g., 14 channels [15] 12 channels [14], 7 channels [85], and 4 channels [86].

A necessary fit for satellite images comes from its large size, in contrast to traditional CNN methods that receive fixed-size inputs and produce a unique classification for the entire image. Therefore, a strategy widely used in the semantic segmentation of remote sensing images is to subdivide it into patches with the same size as the training samples from a sliding window with a step that allows establishing an overlap interval [87]. Image mosaicking considers mathematical operations (usually averaging) in the overlapping areas to avoid frame junction errors [88]. Albuquerque et al. (2020) evaluate the segmentation's accuracy, considering sliding windows with different overlapping strides. Research has also been carried out to evaluate different sliding window sizes [14,18,89]. The frames' fixed size must consider a dimension that allows the general context to perform the classification without a significant increase in computational complexity and CNN parameters. Thus, balancing these two factors is crucial to ensure object detection and computational efficiency. Instance segmentation, where each object in a category has also a unique identification, requires different adjustments in the patch mosaic compared to the semantic segmentation methods, since it is not possible to perform the simple use of an average between the overlapping areas.

For instance segmentation, image labeling requires polygons that delimit each object individually with its bounding box (coordinates) and pixel-wise segmentation mask. This annotation format is more complex, laborious, and requires highly qualified specialists to label more complex information correctly. Thus, a limitation for detecting remote sensing targets is the lack of publicly available data sets suitable for instance segmentation. Many publicly labeled data sets exist for photographic landscape images, such as LabelMe [90], ImageNet [91], PASCAL [92], Cityscapes [93], Open Images [94], and Creating Common Object in Context (COCO) [95]. In this context, the two most popular procedures for annotating objects for computer vision data are COCO and Pascal Visual Object Classes (VOC). Although we do not yet have a large-scale remote sensing image dataset with the appropriate instance segmentation annotations, several databases with raster and vector information can be adapted for this purpose. Therefore, a challenge is to develop a method for converting vector data to the COCO annotation format (data format widely used by instance segmentation and object detection community).

This research aims to perform instance segmentation on multi-channel remote sensing imagery for Center Pivot Irrigation System (CPIS) detection. In this context, the research has three secondary objectives that improve the use of instance segmentation in remote sensing. The first is to develop a method for converting the remote sensing data with its respective vector and raster data to the COCO data format containing the corresponding JavaScript Object Notation (JSON) annotation file. The second is to adapt Detectron2 instance segmentation source code [96] to allow the multispectral data set (the seven surface reflectance bands of Landsat 8 image). Finally, the third is to develop a novel mosaicking method using the sliding window technique and a modified non-max suppression sorted by area to classify large images.

Related Works on Center Pivot Detection

The mapping of CPIS from remote sensing imagery had little changes over time, using predominantly a visual interpretation of circular features since the 70–80 s [97,98] until recently [99–102], with a significant consumption of labor work and time. The different colors, textures, and spectral information inside and between the center pivots make it challenging to obtain accurate classifications by traditional machine learning methods based on pixel or vegetation indices. Consistent automatic detection of center pivots emerges with methods based on deep learning [85,103,104]. Zhang et al. [103] were the precursors in using CNNs for automatic identification of CPIS. The research used an RGB image and did not perform segmentation, and it only identified the central point of each CPIS and established an engagement quadrant with a predetermined size that did not necessarily coincide with the circumference of the central pivot. Subsequently, two articles report the use of semantic segmentation for the detection of CPIS. Saraiva et al. [104] perform the segmentation of the U-Net architecture of the images of the PlanetScope constellation containing four channels (blue, green, red, and near-infrared). De Albuquerque et al. [85] compare three CNN architectures (U-net, Deep ResUnet, and SharpMask) and use Landsat-8 surface reflectance images composed of 7 bands in the rainy and dry period. In this context, instance segmentation is still an unexplored method for this target, which is a differential for the management of irrigated areas, as it establishes the quantity and size of the central pivots, which are fundamental factors for forecasting the harvest and water consumption.

2. Materials and Methods

The present research had the following methodological steps: (a) image data acquisition from three different areas in the rainy and dry period; (b) clipping frames with 512 × 512 pixel dimensions (for the original image and ground truth) with their corresponding annotations in COCO format; (d) data partition into training, development, and test sets (train/dev/test split); (d) training Detectron2 with different backbones; (e) COCO metrics evaluation; and (f) large image mosaicking (Figure 1).

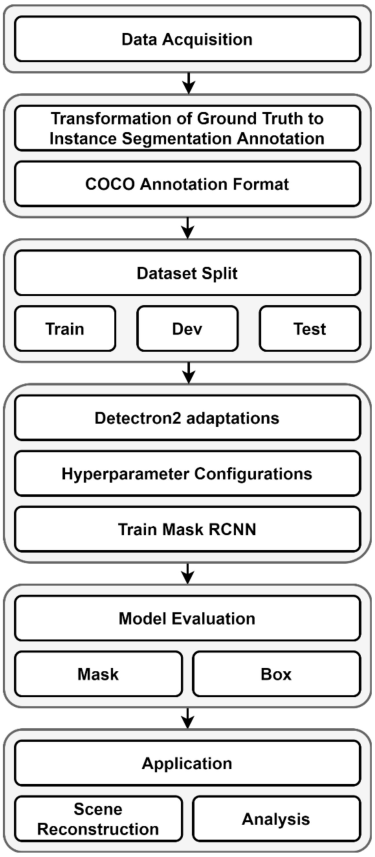
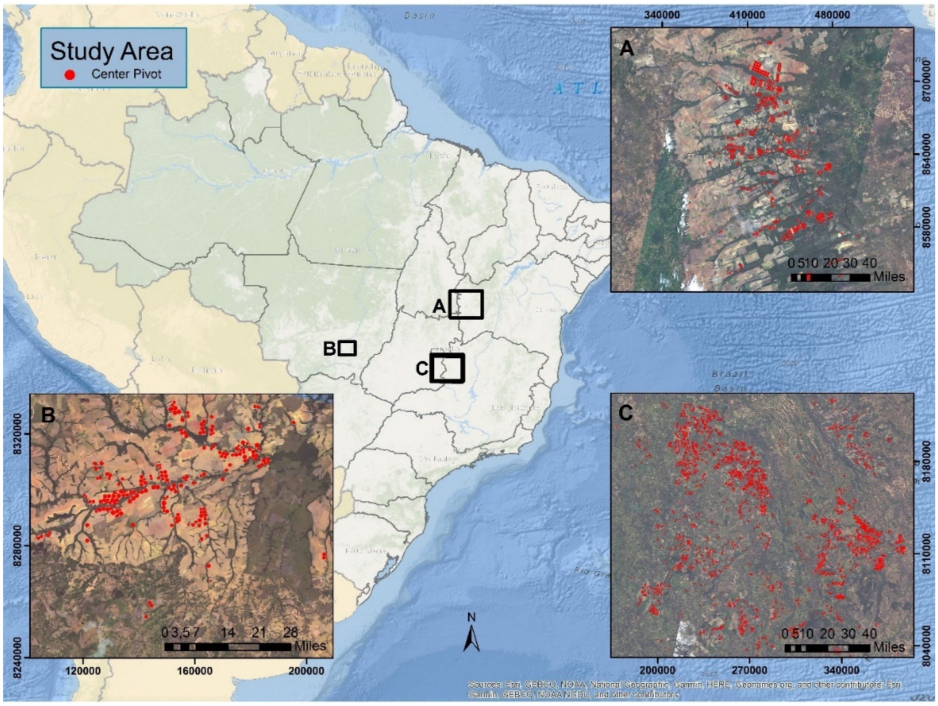


Figure 1. Methodological flowchart of deep instance segmentation of center pivots.

2.1. Dataset and Study Areas

Despite the interest in satellite imagery, few open datasets use multichannel imagery for instance segmentation tasks. The existing datasets are either RGB or for different tasks, such as semantic segmentation or object detection [105–107]. Some open challenges, such as SpaceNet [108] (which provides polygons), could use the same methodology used in this paper to experiment instance segmentation algorithms. Nevertheless, we used the CPIS database developed by Albuquerque et al. [85] based on the survey of center pivots in Brazilian territory by the National Water Agency (ANA) in 2014 [100], since it presents high relevance to agricultural studies. The elaboration of the ANA dataset used visual interpretation on a computer screen. Albuquerque et al. (2020) corrected the data considering the periods of drought and rain for 2015 and 2016 in three regions of Central Brazil. We used surface reflectance images from the Landsat-8 Operational Land Imager (OLI) sensor, containing seven bands and 30-m resolution, for the three regions of Central Brazil. The images correspond to the period of drought and rain.

The study areas locate in the Cerrado biome, presenting a high expansion of center-pivot irrigation due to flat land favorable to mechanization and the dry season between May and September [109]. The three study sites consist of areas around the Federal District, Mato Grosso, and Western Bahia regions, totaling 3731 (more than six thousand considering both seasons) center pivots (Figure 2). The region surrounding the Federal District has the largest number of center pivots in Brazil, not only driven by the proximity of the country's capital but also conflicts over water use [110,111]. In the last decades, the Western Bahia region has presented an advanced agribusiness growth with the expansion of irrigated areas and water conflicts [112–114]. Finally, the state of Mato Grosso has favorable environmental factors for agriculture presenting a 175% growth in CPIS in the period 2010–2017 [99].



Sources: Esri, GEBCO, NOAA, National Geographic, Garmin, HERE, Geonames.org, and other contributors; Esri, Garmin, GEBCO, NOAA NGDC, and other contributors

Figure 2. Location map of the study areas: (**A**) Western Bahia; (**B**) Mato Grosso; and (**C**) surrounding the Federal District.

2.2. COCO Annotation Format

Semantic segmentation algorithms need only a ground truth mask where each element has a class, e.g., pivots (1) and non-pivots (0). Meanwhile, instance segmentation has additional complications in the labeling and annotation format, requiring that each element in a sample image in the training process needs a unique value. For example, an instance segmentation mask with ten center pivots needs different values for each pivot, contrasting with semantic segmentation masks, where all pivots have the same value.

Most of the instance segmentation algorithms follow the COCO annotation format. Thus, we developed a methodology to generate and convert training samples (composed of Landsat images and polygon labels) in the COCO annotation format. This procedure does not aim to replace labeling software, i.e., LabelMe, but to give an alternative for cases in which there is polygon data from the targets, which is common in the remote sensing community. The conversion procedure uses two programs: (a) program developed in this research to extract the samples in frames with a predetermined size and compatible input data with the next program, and (b) Cocosynth repository (https://github.com/akTwelve/cocosynth) [115] with some adaptations that convert the data to the COCO annotation format (Figure 3).

The first program developed in the C++ language considers the following input data: remote sensing image with the respective number of bands, labeled image, vector point of the frame's centroid, and the parameters height and width of each frame. The labeled image is elaborated by converting vector polygons to raster, where each center pivot acquires a distinct integer value from 1 to N, where N is the number of center pivots in the entire scene. The program modifies the labeled image to be compatible with the Cocosynth program that uses different colors for each instance. Thus, the program modifies the polygon identifiers to RGB system values, using an algorithm, like the numerical base conversion (decimal to base-256). The RGB numerical system has 16,777,216 (256 × 256 × 256) color possibilities. The algorithm consists in performing two consecutive divisions by 256. First, the integer number is divided by 256, and the Red color value (R) is the remainder. Consequently, the integer part of the division result is divided by 256 again, where the Green color value (G) is the remainder, and the Blue color value (B) is the integer part of the second division (Equations (1)–(3)). The polygon values start at one instead of zero since the (0,0,0) is the background color. The first integer with value 1 representation is (1,0,0), while the integer 16,777,216 representation is (255,255,255). The color conversion within the image is from left to right and top to bottom direction. Figure 4 shows the processing steps from the polygons to the RGB image. Nevertheless, the program changes the labeled image type (".tiff" file with integer numbers ranging from 1 to the number of instances) to a more straightforward data conversion (".PNG" file with the RGB channels). The proposed C++ program creates a JSON file with each frame information (original image and label data), such as the color, category, and super category of each object.

$$R = \begin{cases} value, & if\ value < 256 \\ remaider\left(\frac{value}{256}\right), & if\ value \geq 256 \end{cases}, \tag{1}$$

$$G = \begin{cases} 0, & if\ value < 256 \\ \left(int\left(\frac{(value)}{256}\right)\right), & if\ 256 \leq value < 65,536 \\ remaider\left(\left(int\left(\frac{value}{256}\right)\right)/256\right), & if\ value \geq 65,536 \end{cases}, \tag{2}$$

$$B = \begin{cases} 0, & if\ value < 65,536 \\ int\left(\frac{value}{65,536}\right), & if\ value \geq 65,536 \end{cases}. \tag{3}$$

The next step to create the COCO annotation file was to adapt the Cocosynth code (coco_json_utils.py) [115] to allow the management of multi-channel remote sensing images in ".tiff" or ".tif" format. This code uses the JSON-file created by our C++ program with color, category, and super-category and creates a new JSON file in the COCO annotation format, which is ready to train.

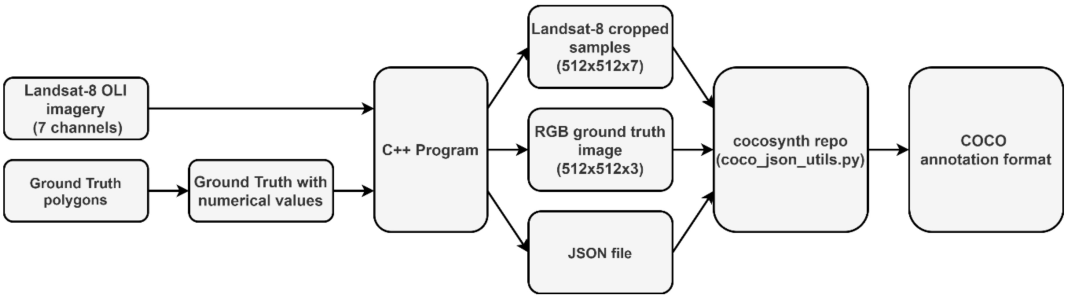

Landsat-8 OLI imagery (7 channels)
Ground Truth polygons
Ground Truth with numerical values
C++ Program
Landsat-8 cropped samples (512x512x7)
RGB ground truth image (512x512x3)
JSON file
cocosynth repo (coco_json_utils.py)
COCO annotation format

Figure 3. Flowchart to obtain samples for training and Creating Common Object in Context (COCO) annotation format.

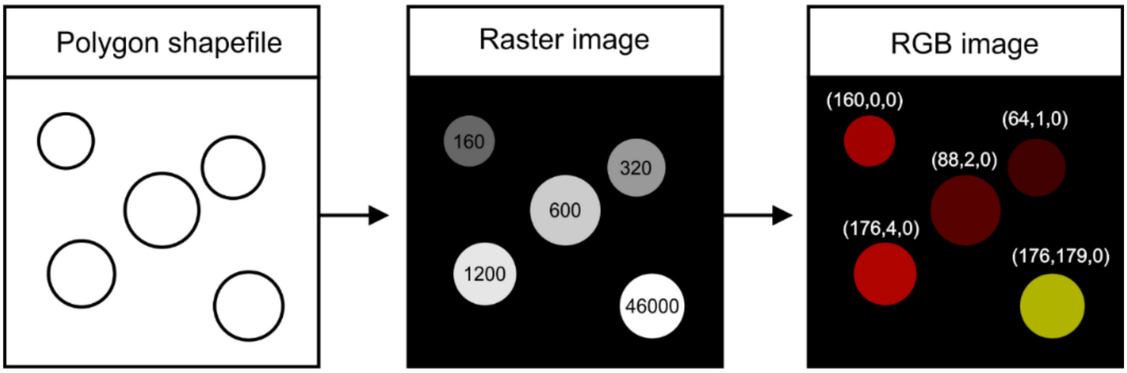


Figure 4. Diagram showing the conversion of data from shapefile to Red, Green, and Blue (RGB) image.

2.3. Data Split

In the scientific literature, there is not a predetermined optimal train/validation/test split. We used 228 images for training and 50 images for test and validation (approximately 70/15/15). The Landsat-8 training images had a 512 × 512-pixel dimension, resulting in 512 × 512 × 7 input shape. The choice of window size considered a larger image size to minimize the edge effects and computational capacity. Table 1 lists the number of instances used in each process. Despite the number of images is not extensive, there is a high concentration of instances, which is the most important number to train the algorithm. In addition, we applied data augmentation considering brightness, contrast and resizing in the training data. This kind of procedure avoids overfitting, and enhances the model ability to learn new features.

Table 1. Image split.

	Number of Images	Number of Instances
Train	228	4762
Validation	50	650
Test	50	850

2.4. Mask R-CNN

One of the most powerful instance segmentation frameworks is the Mask R-CNN [116], introduced by the Facebook Artificial Intelligence Research (FAIR), which combines object detection and semantic segmentation, an evolution of the RCNN [117], Fast RCNN [118], and Faster RCNN [119] methods. This framework operates in two stages: (a) generation of region proposals; and (b) classification of each generated proposals.

We used the Detectron2 [96], a software powered by the Pytorch framework, containing many backbone structures and a faster training process (Figure 5). The original code (https://github.com/facebookresearch/detectron2) uses libraries restricted to RGB in more traditional formats, such as PNG and JPEG formats, whereas satellite images present more channels in the TIFF format. Thus, we implemented changes to read and train multi-channel images in the TIFF format.

2.4.1. Backbone Structure

The input image passes through a convolutional network, also called the backbone structure (Figure 6). The backbone may vary according to the desired tradeoff between performance, training speed, and limitations due to computational power.

The Mask-RCNN architecture consists of a bottom-up and top-down pathway. The bottom-up section is responsible for the convolutions and generation of the feature maps, and the most used structure is the ResNets [120] or ResNeXts [121] with five convolutional modules (C1, C2, C3, C4, and C5). The strides between each module doubles, this means the image dimensions halves. Each convolutional module composition includes many layers that may vary depending on the configurations chosen on the depth of the ResNet. The more layers, the longer it takes to train, but the accuracy tends to be higher, especially in complex object detection. In the present research, we used ResNet50, ResNet101, and ResNeXt101. ResNeXts often present better results when compared to the ResNet since it uses multiple parallel convolutions. Figure 7 shows a simplified structure, where the number of those convolutional blocks in the ResNeXt is the cardinality. Xie et al. [121] tested different cardinality values (1, 2, 4, 8, and 32), showing the best results using 32 (the one used in this research). The input and output dimensions (256d) from the ResNet and ResNeXt are the same, demonstrating similar levels of complexity, varying on the convolutional structures.

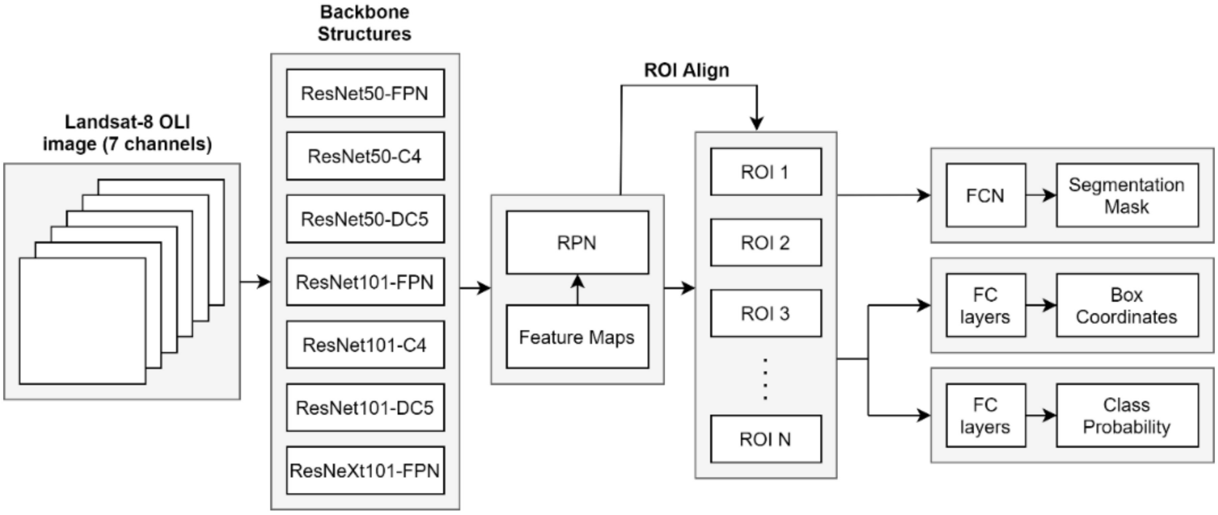


Figure 5. Mask Region-based Convolutional Neural Network (R-CNN) architecture.

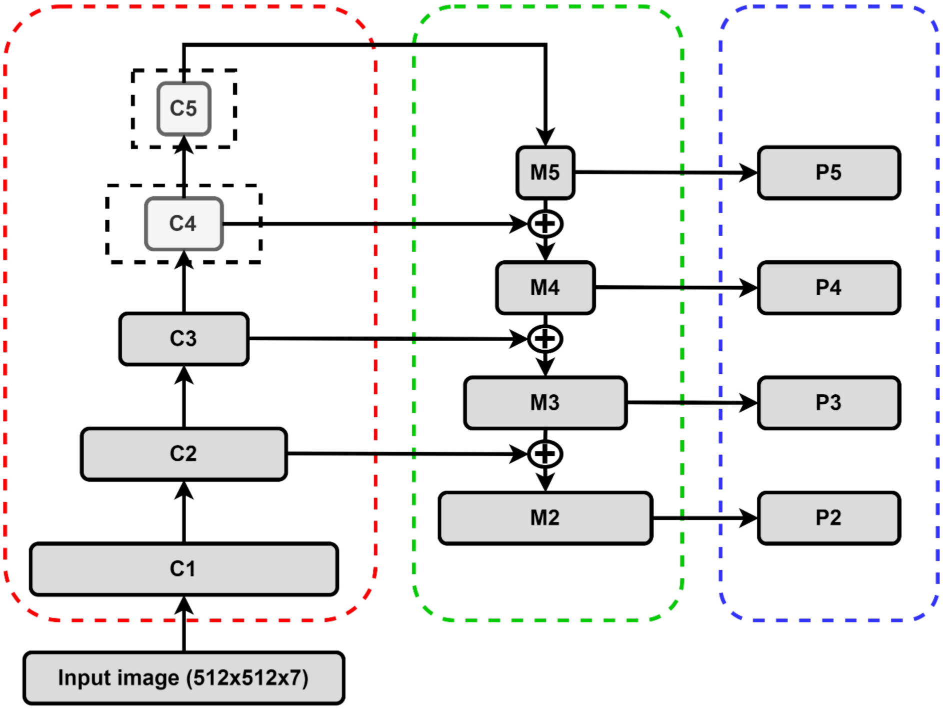


Figure 6. Backbone showing the combination of a ResNet/ResNeXt architecture (red dotted line) with the Feature Pyramid Network (FPN) (green dotted line) and the predictions (blue dotted line).

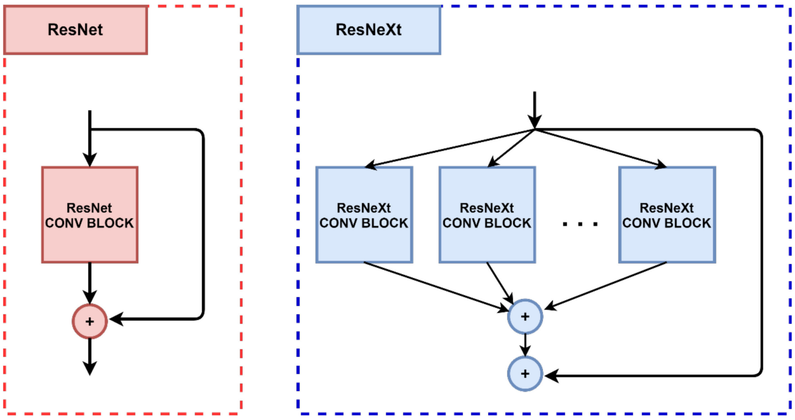


Figure 7. ResNet and ResNeXt configuration (modified from Xie et al. [121]).

The top-down section is a feature extractor with four modules (M5, M4, M3, and M2), where the spatial dimensions double from one module to the other. The higher module (M5) has higher semantic information and smaller spatial dimensions, where the lower modules have a higher spatial dimension and less semantic value. The module stops at M2 instead of M1 because the spatial resolution gets too big, slowing the training process significantly. This top-down structure is often a Feature Pyramid Network (FPN) [122] or a variation. In this present research, we used three different Feature extractors: (a) FPN; (b) Dilated C5; and (c) C4. The Dilated C5 (DC5) multiplies the C5 module by a constant value, altering the dimensionality, where the C4 corresponds to a structure ending in the C4 convolutional module instead of C5.

The bottom-up and top-bottom pathways link through lateral connections, ensuring spatial cohesion from a module to another. In addition, each module in the feature extractor gives a prediction (P5, P4, P3, P2), that will be used in the Region Proposal Networks. The greater the number of convolutional layers, the more complex information the algorithm tends to learn, but also rises the risk of overfitting, and applying dilation on the convolutional modules may increase performance on different sized objects. Thus, testing different structures is essential to obtain optimal results. We compared seven different backbone structures (ResNet50-FPN, ResNet50-DC5, Resnet50-C4, ResNet101-FPN, ResNet101-DC5, ResNet101-C4, and ResNeXt101-FPN).

2.4.2. Region Proposal Network and Region of Interest (ROI) Align

The backbone output (P2, P3, P4, P5) are feature maps used in the Region Proposal Network (RPN) to generate anchor boxes. Each region with high probability generates 9 anchor boxes with different ratios (1:1, 2:1, 1:2) and scales (0.5, 1, 2). The Region of Interest (ROI) pass through ROI align (Figure 8), a bilinear interpolation quantization-free o preserves spatial information (He et al., 2016). These fixed dimension ROIs enter three parallel processes: (a) class of the object and its respective probability; (b) bounding box; and (c) segmentation mask.

2.4.3. Loss Functions

The total loss of the training process is the addition of mask loss, class loss, and box regression (Equation (4)). The segmentation mask is a binary classification that involves a single classifier per class (one versus all strategy). Therefore, each ROI will only consider one object at a time. Thus, the loss function is a simple log loss [118], in which the result is the average from all results (Equation (5)). The classification loss is also the same formula.

There are two ways to obtain the bounding box, considering the four coordinate values: (a) using “x” the centroid in the x-axis; “y” the centroid in the y-axis; (h) the height of the box; and “w” the width of the box boxes [123]; and (b) using: “x1” the minimum x value; “x2” the maximum x value; “y1” the minimum y value; and “y2” the maximum y value. The Detectron2 algorithm uses the second method, and its loss regression function uses *L1 loss* (Equation (6)). Figure 9 shows the process after a loss reduction from the first to the second iteration. The computed loss is lower in the second iteration because the differences are smaller between ground truth (black dotted line) and the prediction (red line).

$$L = L_{mask} + L_{cls} + L_{BBox} \,, \tag{4}$$

$$L_{mask} \; and \; L_{cls} \; = \sum_{i=1}^{N} y_i * \log(p(y_i)) + (1 - y_i) * \log(1 - p(y_i)) \,, \tag{5}$$

$$L1 \; Loss = \sum_{i=1}^{N} |y_{true} - y_{predicted}| \,. \tag{6}$$

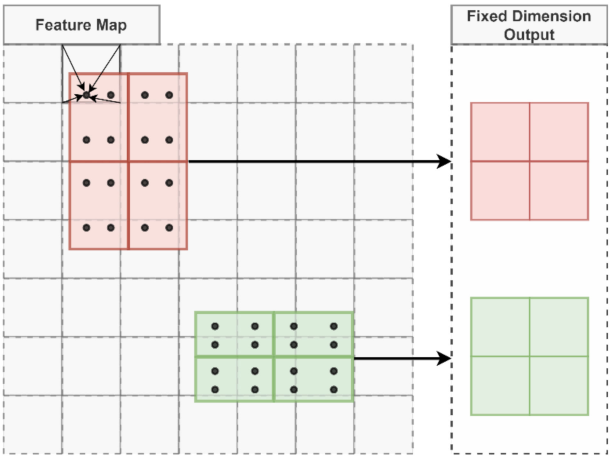


Figure 8. Region of Interest (ROI) align method (modified from He et al. 2016).

2.4.4. Hyperparameter Configuration

Another critical step in training a neural network is the hyperparameter configuration. Thus, we trained from scratch (unfreezing all layers) seven models using all seven channels and the best model using only the RGB channels (Landsat-8 bands 2, 3, and 4). We used: (a) Adam optimizer with a learning rate of 0.001 divided by ten after 1000 iterations and momentum of 0.9; (b) 256 ROIs per image; (c) 30,000 iterations, keeping track of the validation loss to an optimal converging point and avoid overfitting; (d) five anchor boxes sizes of 16, 32, 64, 128, 256; (e) 1000 warm-up iterations (where learning rate slowly increases to avoid errors) with a 0.001 factor; and (f) 1 image per batch. In addition, we used Nvidia GeForce RTX 2080 TI GPU with 11 GB memory.

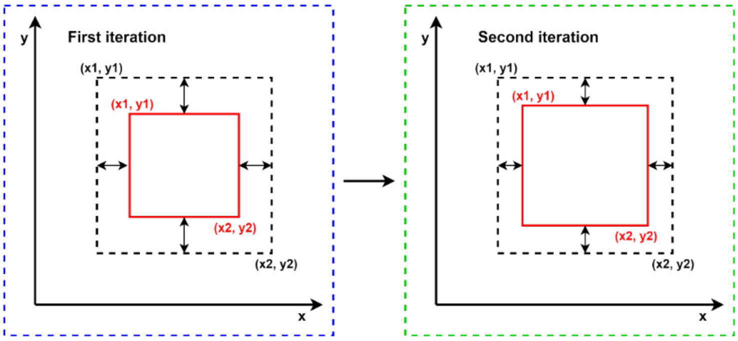


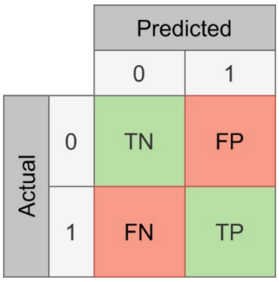
Figure 9. Bounding Box, where the red line represents the ground truth, and the dotted black line is the predicted bounding box.

Data normalization (z-score method) was necessary since each channel has different ranges of values and can bring bad results during the training process, such as disappearance gradients [124] (Equation (7)). Furthermore, normalization allows us to accelerate the training process.

$$x' = \frac{x - average(x)}{std(x)} . \tag{7}$$

2.5. Accuracy Analysis

Accuracy analysis is crucial in Deep Learning tasks to evaluate how well the trained model behaves in new data, which is a powerful insight to understand applicability in the real world. The confusion matrix shows each class's frequencies, being extremely useful to evaluate the supervised models of Machine Learning/Deep Learning. Figure 10 shows the confusion matrix, where True Positives (*TP*) and True Negatives (*TN*) represent elements correctly identified in their corresponding classes. In contrast, False Positives (*FP*) and False Negatives (*FN*) represent misclassified elements.



		Predicted	
		0	1
Actual	0	TN	FP
	1	FN	TP

Figure 10. Confusion matrix.

The two-primary metrics for evaluating instance segmentation models are precision (Equation (8)) and recall (Equation (9)). Precision is the number of correctly identified positive instances (*TP*) divided by the total number of predictions (*TP* + *FP*), and recall is the number of correctly identified positive instances divided by the total number of positive instances (*TP* + *FN*).

$$Precision = \frac{TP}{TP + FP} , \tag{8}$$

$$Recall = \frac{TP}{TP + FN} . \tag{9}$$

Precision and recall bring rich insights to data, but, when dealing with deep learning algorithms, the results are often probabilities, and another crucial information is the threshold cutoff point. The threshold considers the Intersection over Union (IoU) of the bounding boxes (Figure 11). A low IoU will be more permissive when considering possible targets, and a large IoU will be more restricted. The optimal point may vary depending on each problem.

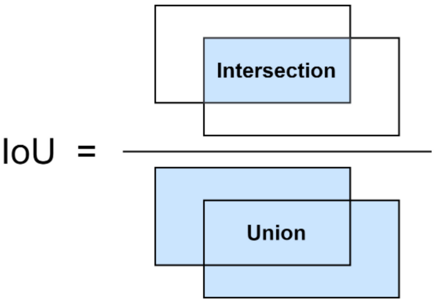


Figure 11. Intersection over Union (IoU) visual representation.

The instance segmentation model's evaluation used the standard COCO metrics, including (a) Average Precision (AP); (b) AP_{50}; (c) AP_{75}; (d) AP_{small}; (e) AP_{medium}; and (f) AP_{large}, (g) Average Recall with 100 maximum detections (AR_{100}) [95]. These are the most commonly used metrics in instance segmentation tasks, proving to be satisfactory to evaluate and compare different models in object detection and segmentation (mask quality) performance, including the original Mask RCNN research [56] You Only Look At Coefficients (YOLACT) [125], YOLACT++ [126], mask scoring RCNN [67], and cascade RCNN [66], among other works using applications of these methods.

The average precision (AP) uses the mean value from 10 IoU thresholds, starting at 0.5 up to 0.95 with 0.05 steps (0.50: 0.05: 0.95). The closer the AP is to 1, the better the model. AP_{50} represents the calculation under the IoU threshold of 0.50, whereas AP_{75} is a stricter metric and represents the calculation under the IoU threshold of 0.75. In addition, the metrics consider the average precision in different target sizes, having three categories (a) small (area < 32^2 pixels), (b) medium (32^2 pixels < area < 96^2 pixels); and (c) large (area > 96^2 pixels). The present research does not have objects larger than 96^2 pixels; thus, we will only consider AP_{small} and AP_{medium}. Another important metric is the Average Recall (AR), where the averaged IoU thresholds are the same from the AP (0.50: 0.05: 0.95). Furthermore, the AR considers the maximum number of detections (Max Dets). Since the maximum number of detections in a single 512 × 512-pixel frame in our dataset is 96, we will only consider the AR with a maximum detection of 100 objects (AR_{100}). Other options analyzed in the COCO dataset is considering 1 and 10 detections, which would not bring much value to the observations.

2.6. Scene Mosaicking

Remote sensing images are larger than the image size used for training and validation due to computational limitations. For example, the center pivot survey covers a wide area, not restricted to just a single frame. Therefore, the classification of a complete scene requires a mosaic reconstruction of sub-images with training image size. For this reason, we used the sliding window technique that runs through the image with a specific dimension (height × width) and a stride value in the horizontal and vertical directions. When the stride is smaller than the window size, it creates an overlap between consecutive frames. Semantic and instance segmentation errors occur predominantly at the frame edges, corrected, or minimized with overlapping images [85,87].

The sliding window with a stride dimension corresponding to half-frame length shows three patterns (Figure 12): (a) base arrangement (initial position at x = 0 and y = 0) (Figure 12A); (b) horizontal displacement arrangement (initial position at x = half-frame length and y = 0) (Figure 12B); and (c) vertical displacement arrangement (initial position at x = 0 and y = half-frame length) (Figure 12C).

In this configuration, window overlays guarantee three classifications for the same object (disregarding an edge with half-window length). Incomplete classifications at the window edges (red and orange boxes) should be eliminated (Figure 13A), remaining in these places only the boxes (marked in green) from the two other arrangements (horizontal or vertical) (Figure 13B). We restricted the valid boxes to the central zone of the vertical and horizontal displacement arrangements where edge errors concentrate on the base arrangements, optimizing and eliminating information redundancy. Figure 13 shows the green boxes as the appropriate result of the conjunction of the base (Figure 13A) and horizontal and vertical configurations (Figure 13B).

The bounding box position of a given sliding window is repositioned to a coordinate system that considers the entire image. Consequently, data processing is windowed, but storage considers the size of the original image. Besides, each object's description uses a binary mask with the total dimension of the image (filled with zeros). Therefore, each new element store uses a new dimension of the array with shape (Number of instances, width, height). We store four types of information in a NumPy array: (1) bounding box coordinates (N, x1, x2, x3, x4); (2) class labels for each bounding box (N, classification); (3)

prediction for each bounding box (N, predictions); and (4) prediction masks for each frame (N, image height, image width).

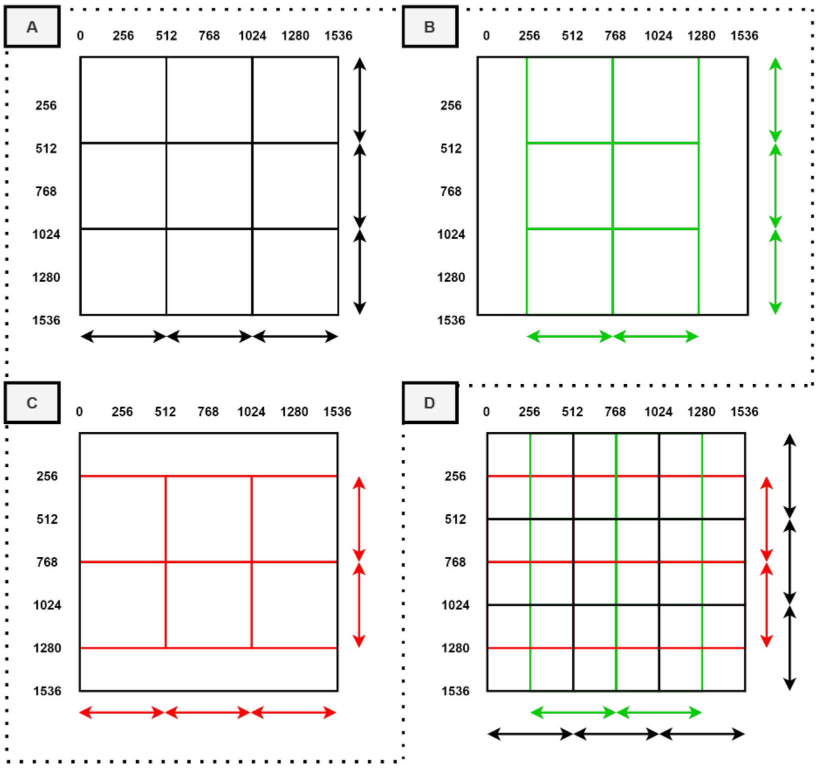


Figure 12. Visual representation of (**A**) base arrangement, (**B**) horizontal displacement arrangement, (**C**) vertical displacement arrangement, and (**D**) the combination of (**A**), (**B**), and (**C**).

To exclude excessive bounding boxes, we apply a modified no-maximum suppression algorithm that uses the box size and the overlapping area index. The method calculates the bounding box area by its coordinate pairs in the upper left corner ($x1, y1$) and lower right corner ($x2, y2$) (Equation (10)), sort by size, and select the largest. The elimination of the boxes is from the smallest to the most extensive areas to avoid possible errors.

$$Box\ Area = (x2 - x1) * (y2 - y1). \quad (10)$$

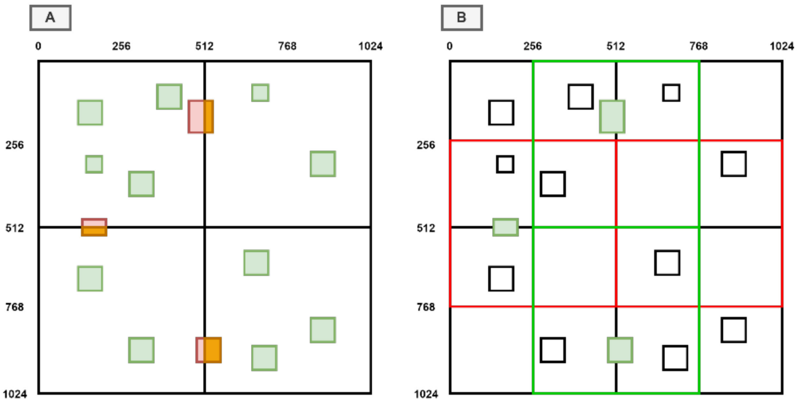


Figure 13. Theoretical representation of objects in the frame edges, with partial classifications from the base classifier (**A**) and complete classifications from the horizontal and vertical arrangements (**B**).

To ensure that we are eliminating overlapping boxes, we use a ratio that is the total overlap area divided by the box area (Equation (11)), considering the Overlap Box Width

(*OBW*) and Overlap Box Height (*OBH*) (Equations (12) and (13)) (Figure 14). The coordinate values increase from top to bottom and from left to right. We consider an overlap of 0.3 to exclude excessive boxes (keeping the box with the largest area).

$$Overlapping\ Ratio = \frac{OBW * OBH}{Box\ Area}, \tag{11}$$

$$OBW = \max(B1(x1); B2(x1)) - \min(B1(x2); B2(x2)), \tag{12}$$

$$OBH = \max(B1(y1); B2(y1)) - \min(B1(y2); B2(y2)). \tag{13}$$

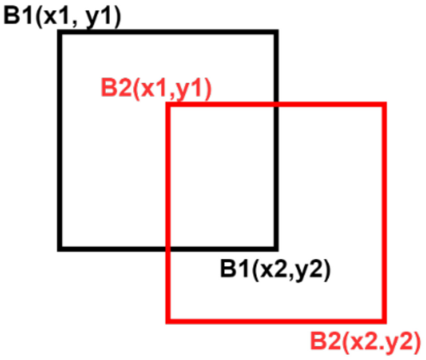


Figure 14. Demonstration of the bounding box coordinates.

Figure 15 shows three boxes for the same object. The red and orange boxes are at the edges of two consecutive frames, classifying only parts of the object, while the green box classifies the entire object. The ordering by area (Figure 15A) guarantees the elimination of smaller frames (partial target). In the present case, the procedure becomes more appropriate than the ranking by score (Figure 15B), which selects the highest confidence score, since it is not always the box that maps the entire object.

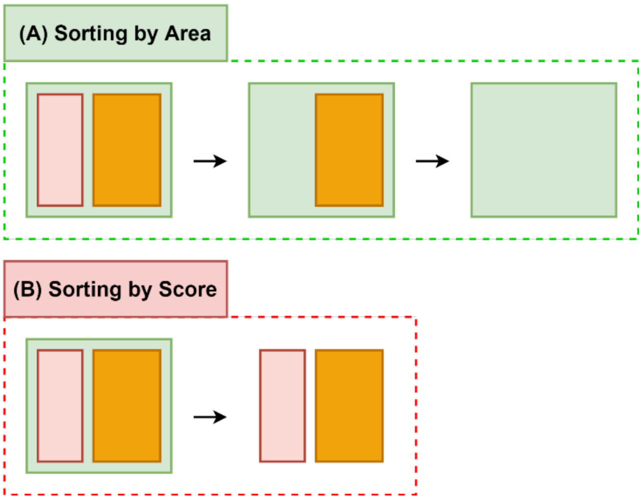


Figure 15. Demonstration of sorting by area (**A**) and a possible scenario of sorting by score (**B**).

3. Results

3.1. Ground Truth COCO Transformation

Figure 16 shows an example of a 512 × 512-pixel frame before and after running the program that converts the polygon identifiers to the RGB system and creates the JSON format file with the annotation information. This procedure allows an easy transformation to the COCO annotation system used in the training phase.

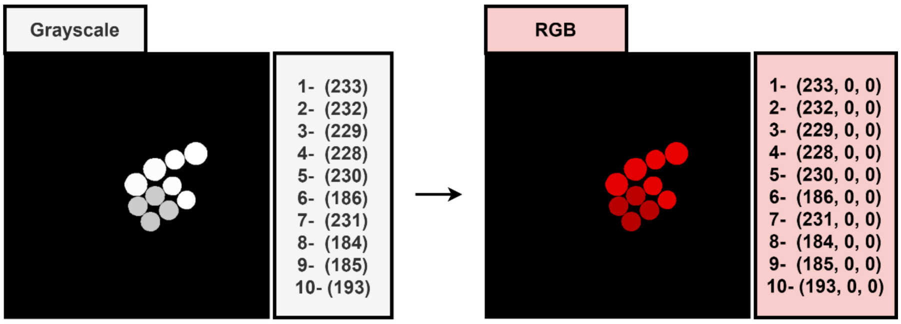


Figure 16. Representation of new ground truth annotated data.

3.2. *Evaluation of COCO Metrics*

Tables 2 and 3 list the COCO metrics, for instance segmentation. ResneXt101 presented the best results, followed by Resnet101-FPN. The backbone structures from 50 to 101 depth in the Resnet architecture show significant differences in almost 10% average precision. The ResneXt101 has similar results to ResNet101 when analyzing the Average Precision (AP) with the IoU threshold at 0.5. However, the difference is significant at IoU 0.75, with nearly 2% improvement compared to the second-best model (Resnet101-FPN). Medium-sized CPIS detection is also greater than smaller ones.

Another crucial analysis is the performance comparison using multi-channel imagery considering seven channels with the traditional RGB images (Landsat-8 bands 2, 3, 4). Thus, we applied the best model (ResNext101-FPN) using the same train/dev/test images but considering only the RGB channels. Results show a strong tendency of accuracy advantages using more channel information, demonstrating that the usage of multi-channel imagery, especially to remote sensing data, where the tradeoff between accuracy and processing speed in most cases tilts toward accuracy.

Table 2. Metrics precision (7 channels).

Backbone	Type	AP	AP_{50}	AP_{75}	AP_{small}	AP_{medium}	AR_{100}
Resnet50-FPN	Mask	70.567	86.095	81.150	56.214	77.494	75.2
	Box	69.142	86.425	82.452	57.154	78.110	74.7
Resnet50-DC5	Mask	65.28	81.722	79.185	43.874	75.344	72.3
	Box	63.017	82.435	80.746	48.541	72.554	70.4
Resnet50-C4	Mask	67.835	82.334	82.294	50.233	78.400	73.1
	Box	65.561	83.390	81.162	49.392	74.963	70.9
Resnet101-FPN	Mask	75.213	90.915	87.601	64.564	83.047	80.6
	Box	74.415	91.618	87.806	64.715	80.978	80.1
Renset101-DC5	Mask	74.408	90.542	86.151	62.163	82.615	78.8
	Box	73.624	90.343	86.390	62.421	81.037	78.6
Resnet101-C4	Mask	74.776	90.765	86.611	62.665	83.370	79.0
	Box	73.814	90.473	86.868	62.846	81.161	78.9
ResneXt101-FPN	Mask	77.970	93.758	90.620	67.585	84.776	82.3
	Box	77.433	93.651	90.459	68.545	82.933	82.1

Table 3. Mask precision (RGB).

Backbone	Type	AP	AP_{50}	AP_{75}	AP_{small}	AP_{medium}	AR_{100}
ResneXt101-FPN	Mask	74.776	92.417	87.605	64.619	81.781	78.8
	Box	74.562	92.928	88.506	65.947	80.394	78.4

3.3. *Image Mosaicking*

The process of creating the bounding boxes and segmentation co-occur. Nevertheless, to give a better visual understanding, Figure 17a shows the results from the base classifier (stating at x = 0 and y = 0 with 512-pixel step), which outputs a classification to all objects.

Figure 17b shows the classification of the horizonal classifier (starting at x = 256 and y = 0 with 512-pixel step), considering only center pivots that start before the center of the image (x < 256) and ends after the center of the image (x > 256). Figure 17c shows the deleted boxes in the non-max suppression sorted by area algorithm, evidencing the correct elimination from the partial classification in Figure 17a. Finally, Figure 17d shows the final classification from this small example, where only the correct boxes remain, and there is only one classification per object, demonstrating the effectiveness of the algorithm.

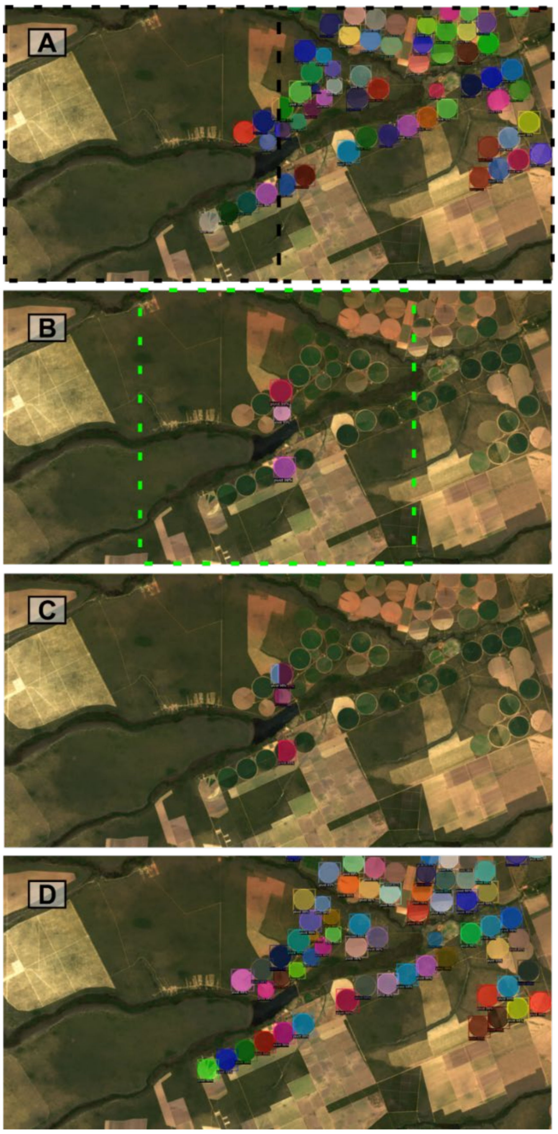


Figure 17. (**A**) Represents a full classification, (**B**) represents the horizontal edge classification, considering only elements in the middle, (**C**) represents the deleted boxes, and (**D**) represents the final classification.

The same procedure applied to an entire applying the non-max suppression sorted by area result in the classified image (Figure 18). Other information we can extract immediately is the number of objects and the average size of a center pivot in the referred region. This kind of information is vital to public managers and farmers to understand its plantation and surroundings.

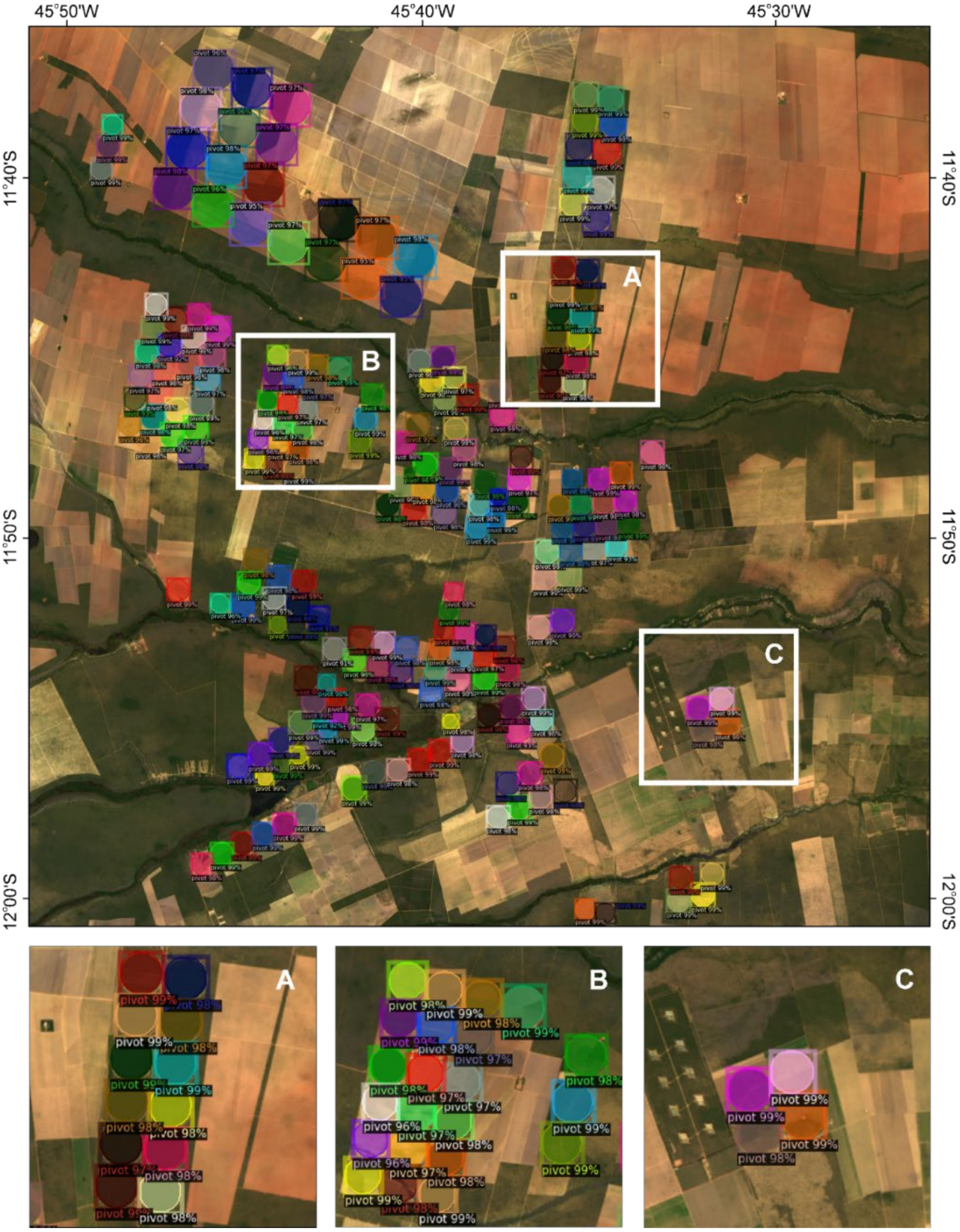


Figure 18. This figure shows 1536 × 1536 classified image using the proposed mosaicking method, and three zoomed areas: (**A**), (**B**) and (**C**).

4. Discussion

This research presents the results of state-of-the-art instance segmentation (Mask-RCNN) in satellite images with an innovative approach that uses large and multi-channel images. Instance segmentation brings more information than semantic segmentation, enabling a greater understanding of the scenes. The box boundaries and mask predictions better visualize different instances and enable useful insights, such as object coordinates, number of instances, average object size, total area occupied, and powerful to remote sensing tasks.

There are currently no works using Mask-RCNN algorithms in multichannel imagery. Previous works on object detection using multi-channel imagery use segmentation-first strategy (object-based Convolutional Neural Networks) [86,127,128]. The limitation of the segmentation-first strategy is that objects receive the same semantic information for all instances. In contrast, the Mask-RCNN makes a clear distinction between objects and gives per-object information, showing promising results even when objects overlap [72]. Therefore, the instance segmentation in the remote sensing data predominantly uses the Mask-RCNN/Faster-RCNN architecture [129–132]. However, the instance segmentation in remote sensing has been limited to RGB channels or even one channel of the Sentinel-1 image. In this way, for the best of our knowledge, the present research was the first to use Mask-RCNN with remote sensing multi-channel, demonstrating that this information increases performance and target detection.

Considering the instance segmentation in RGB images, researches with Mask-RCNN obtained relevant accuracy. Su et al. [130] applied ResNet50-FPN and ResNet101-FPN backbones in a Mask-RCNN and a proposed new method changing the pooling technique in NWPU Very High Resolution (VHR)-10 dataset, which contains mostly RGB imagery and a few pan-sharpened color infrared images. The authors evaluated the COCO metrics (AP, AP50, and AP75). The best model had results (64.8 AP, 93.8 AP_{50}, and 73.2 AP_{75} mask results and 61.2 AP, 94 AP_{50}, and 72.1 AP_{75} detection results) similar to our (77.970 AP, 93.758 AP_{50}, 90.620 AP_{75} mask results and 77.433 AP, 93.651 AP_{50}, and 90.459 AP_{75} detection results), demonstrating that instance segmentation models in remote sensing imagery targets present high accuracy. Zhao et al. [133] applied a boundary regularization for building extraction using the Mask-RCNN algorithm and ResNet101-FPN as the backbone structure. The authors used the COCO annotations format and compared the proposed method with the traditional Mask-RCNN models using the F1 score metrics. The Mask-RCNN outperformed their algorithm. Yekeen et al. [77] applied Mask-RCNN in oil spill detection using Keras and Tensorflow and ResNet101-FPN backbone in Synthetic-Aperture Radar (SAR) imagery. The authors analyzed precision, recall, specificity, f1, IoU, and overall accuracy, showing promising results. Despite the good results, the usage of the Detectron2 algorithm (which contains more backbone structures) would increase performance using the ResNeXt101 architecture.

In this research, the instance segmentation of large images used a mosaic of overlapping frames from sliding windows with non-maximum suppression by area index. The current approach is essential for remote sensing images that predominantly have more significant dimensions. The large image reconstruction from the sliding window mosaic is widely used in the literature for semantic segmentation [85] propose a sliding window technique for semantic segmentation to minimize border effects. To show these metrics, they monitored the Area Under the Receiver Operating Characteristic (ROC) curve to measure the increasing performance, demonstrating a powerful tool for semantic segmentation scene mosaicking. Similarly, Yi et al. [87] applied scene reconstruction in a semantic segmentation algorithm for building extraction training with 256 × 256 pixel patches and mosaicking with a sliding window with a 64-pixel stride to minimize errors. Nevertheless, these solutions are not applicable to object detection, where each instance has a bounding box and different values. Martins et al. [86] applied a segmentation-first strategy algorithm in multi-channel National Agriculture Imagery Progam (NAIP) imagery (four channels) to classify large scenes. They used different patch sizes in the convolutional neural networks training process to better predict different sized data. In our work, the Mask-RCNN algorithm uses different anchor boxes that do this job very efficiently, especially when using deep backbone structures, such as ResNeXt101-FPN. In addition, the instance-first strategy, where each object has a unique mask segmentation, gives better results when there are overlapping objects, which is very common in object detection.

5. Conclusions

Instance level recognition, which requires individual objects' limits, allows a more thorough understanding of the image content with high potential for remote sensing. Instance segmentation is exceptionally suitable for different applications essential to counting different objects and estimating its areas individually. This research used the Detectron2 algorithm, the current state-of-the-art in instance segmentation, and still with little exploration in satellite images. The present research innovates in the following aspects: (a) development of a method to convert vector polygons from the interpretation of remote sensing images to the COCO format with its JSON file; (b) adaptation of the Detectron2 algorithm for multi-channel processing, and (c) proposition of a method for processing large images considering sliding windows and mosaic reconstruction by non-maximum suppression. The novel approach, in instance segmentation using the sliding window technique, gives a more substantial analysis since it is possible to gather information in large images.

This study applied the developed methodology for CPIS detection, which is a vital aspect of the support system of agricultural management and water resources. The detection of CPIS is a challenging task due to the different and complex crop patterns. Previous surveys have applied manual methods, and, only recently, semantic segmentation methods have been used for automatic detection. However, the semantic reserve has limitations for the individual detection of areas, especially as areas are in contact or overlap. We compared seven backbone structures in the Mask-RCNN model (Resnet50-FPN, Resnet50-DC5, Resnet50-C4, Resnet101-FPN, Renset101-DC5, Resnet101-C4, ResneXt101-FPN). In the ResNet50 and ResNet101, the FPN feature extractor outperformed C4 and Dilated C5. In addition, the detection of medium objects is significantly better, with an APmedium nearly 20% higher than the APsmall. The ResNeXt101-FPN is considerably better than the other models with an AP 3% higher than Resnet101-FPN (the second-best model).

Furthermore, a critical conclusion is also the difference between training with RGB and multi-channel imagery. Thus, we compared the best model (ResNeXt101-FPN) training with the same samples but considering only the RGB bands (2, 3, and 4). Results show that using multi-channel imagery improves the accuracy metrics for nearly 3%, evidencing an excellent tendency to other researchers to use multi-channel imagery to improve accuracy.

The proposed methodology improves remote sensing images and applies to studies previously carried out with semantic segmentation. Future work may include creating new backbone structures and small arrangements to allow the instance segmentation for multiclass problems. Besides, the present method applies in other science fields, which use larger images or a more significant number of channels, such as biomedical images. In addition, an extensive database of CPIS data can be developed for model training to provide better results in transfer learning.

Author Contributions: Conceptualization, O.L.F.d.C., O.A.d.C.J., A.O.d.A., and P.P.d.B.; methodology, O.L.F.d.C., A.O.d.A., O.A.d.C.J., P.P.d.B.; software, O.L.F.d.C., C.R.S., P.H.G.F., P.P.d.B., D.L.B.; validation, O.L.F.C., A.O.d.A., C.R.S., and P.H.G.F.; formal analysis, O.L.F.d.C., O.A.d.C.J., P.H.G.F., R.F.G.; investigation, R.F.G., P.H.G.F.; resources, O.A.d.C.J., R.A.T.G., R.F.G.; data curation, O.L.F.d.C., A.O.d.A., P.H.G.F.; writing—original draft preparation, O.L.F.d.C., O.A.d.C.J.; writing—review and editing, O.L.F.d.C., O.A.d.C.J., R.d.S.d.M., D.L.B.; supervision, O.A.d.C.J., R.A.T.G., R.F.G.; project administration, O.L.F.d.C., O.A.d.C.J., R.A.T.G., R.F.G.; funding acquisition, O.A.d.C.J., R.A.T.G., R.F.G. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the following institutions: National Council for Scientific and Technological Development (434838/2018-7), Coordination for the Improvement of Higher Education Personnel and the Union Heritage Secretariat of the Ministry of Economy.

Acknowledgments: The authors are grateful for financial support from CNPq fellowship (Osmar Abílio de Carvalho Júnior, Renato Fontes Guimarães, and Roberto Arnaldo Trancoso Gomes). Special thanks are given to the research group of the Laboratory of Spatial Information System of the University of Brasilia for technical support. The authors thank the researchers form the Union Heritage Secretariat of the Ministry of Economy, who encouraged research with deep learning. Finally, the authors acknowledge the contribution of anonymous reviewers.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Zhu, X.X.; Tuia, D.; Mou, L.; Xia, G.; Zhang, L.; Xu, F.; Fraundorfer, F. Deep Learning in Remote Sensing: A Comprehensive Review and List of Resources. *IEEE Geosci. Remote Sens. Mag.* **2017**, *5*, 8–36. [CrossRef]
2. Liu, L.; Ouyang, W.; Wang, X.; Fieguth, P.; Chen, J.; Liu, X.; Pietikäinen, M. Deep Learning for Generic Object Detection: A Survey. *Int. J. Comput. Vis.* **2020**, *128*, 261–318. [CrossRef]
3. Zhang, L.; Zhang, L.; Du, B. Deep learning for remote sensing data: A technical tutorial on the state of the art. *IEEE Geosci. Remote Sens. Mag.* **2016**, *4*, 22–40. [CrossRef]
4. Nogueira, K.; Penatti, O.A.B.; dos Santos, J.A. Towards better exploiting convolutional neural networks for remote sensing scene classification. *Pattern Recognit.* **2017**, *61*, 539–556. [CrossRef]
5. Ma, L.; Liu, Y.; Zhang, X.; Ye, Y.; Yin, G.; Johnson, B.A. Deep learning in remote sensing applications: A meta-analysis and review. *ISPRS J. Photogramm. Remote Sens.* **2019**, *152*, 166–177. [CrossRef]
6. Zhang, L.; Li, W.; Shen, L.; Lei, D. Multilevel dense neural network for pan-sharpening. *Int. J. Remote Sens.* **2020**, *41*, 7201–7216. [CrossRef]
7. Ma, J.; Yu, W.; Chen, C.; Liang, P.; Guo, X.; Jiang, J. Pan-GAN: An unsupervised pan-sharpening method for remote sensing image fusion. *Inf. Fusion* **2020**, *62*, 110–120. [CrossRef]
8. Liu, L.; Wang, J.; Zhang, E.; Li, B.; Zhu, X.; Zhang, Y.; Peng, J. Shallow-Deep Convolutional Network and Spectral-Discrimination-Based Detail Injection for Multispectral Imagery Pan-Sharpening. *IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens.* **2020**, *13*, 1772–1783. [CrossRef]
9. Liu, X.; Liu, Q.; Wang, Y. Remote sensing image fusion based on two-stream fusion network. *Inf. Fusion* **2020**, *55*, 1–15. [CrossRef]
10. Hughes, L.H.; Schmitt, M.; Zhu, X.X. Mining hard negative samples for SAR-optical image matching using generative adversarial networks. *Remote Sens.* **2018**, *10*, 1552. [CrossRef]
11. Merkle, N.; Luo, W.; Auer, S.; Müller, R.; Urtasun, R. Exploiting deep matching and SAR data for the geo-localization accuracy improvement of optical satellite images. *Remote Sens.* **2017**, *9*, 586. [CrossRef]
12. Wang, S.; Quan, D.; Liang, X.; Ning, M.; Guo, Y.; Jiao, L. A deep learning framework for remote sensing image registration. *ISPRS J. Photogramm. Remote Sens.* **2018**, *145*, 148–164. [CrossRef]
13. Ye, F.; Xiao, H.; Zhao, X.; Dong, M.; Luo, W.; Min, W. Remote Sensing Image Retrieval Using Convolutional Neural Network Features and Weighted Distance. *IEEE Geosci. Remote Sens. Lett.* **2018**, *15*, 1535–1539. [CrossRef]
14. De Bem, P.P.; de Carvalho Júnior, O.A.; de Carvalho, O.L.F.; Gomes, R.A.T.; Fontes Guimarães, R. Performance Analysis of Deep Convolutional Autoencoders with Different Patch Sizes for Change Detection from Burnt Areas. *Remote Sens.* **2020**, *12*, 2576. [CrossRef]
15. De Bem, P.P.; de Carvalho Junior, O.; Fontes Guimarães, R.; Trancoso Gomes, R. Change Detection of Deforestation in the Brazilian Amazon Using Landsat Data and Convolutional Neural Networks. *Remote Sens.* **2020**, *12*, 901. [CrossRef]
16. Zhang, P.; Gong, M.; Su, L.; Liu, J.; Li, Z. Change detection based on deep feature representation and mapping transformation for multi-spatial-resolution remote sensing images. *ISPRS J. Photogramm. Remote Sens.* **2016**, *116*, 24–41. [CrossRef]
17. Peng, D.; Zhang, Y.; Guan, H. End-to-End Change Detection for High Resolution Satellite Images Using Improved UNet++. *Remote Sens.* **2019**, *11*, 1382. [CrossRef]
18. Ammour, N.; Alhichri, H.; Bazi, Y.; Benjdira, B.; Alajlan, N.; Zuair, M. Deep Learning Approach for Car Detection in UAV Imagery. *Remote Sens.* **2017**, *9*, 312. [CrossRef]
19. Chen, Y.; Li, Y.; Wang, J.; Chen, W.; Zhang, X. Remote Sensing Image Ship Detection under Complex Sea Conditions Based on Deep Semantic Segmentation. *Remote Sens.* **2020**, *12*, 625. [CrossRef]
20. Dong, Z.; Lin, B. Learning a robust CNN-based rotation insensitive model for ship detection in VHR remote sensing images. *Int. J. Remote Sens.* **2020**, *41*, 3614–3626. [CrossRef]
21. Yu, Y.; Gu, T.; Guan, H.; Li, D.; Jin, S. Vehicle Detection from High-Resolution Remote Sensing Imagery Using Convolutional Capsule Networks. *IEEE Geosci. Remote Sens. Lett.* **2019**, *16*, 1894–1898. [CrossRef]
22. Diakogiannis, F.I.; Waldner, F.; Caccetta, P.; Wu, C. ResUNet-a: A deep learning framework for semantic segmentation of remotely sensed data. *ISPRS J. Photogramm. Remote Sens.* **2020**, *162*, 94–114. [CrossRef]
23. Volpi, M.; Tuia, D. Deep multi-task learning for a geographically-regularized semantic segmentation of aerial images. *ISPRS J. Photogramm. Remote Sens.* **2018**, *144*, 48–60. [CrossRef]
24. Zhao, W.; Du, S.; Wang, Q.; Emery, W.J. Contextually guided very-high-resolution imagery classification with semantic segments. *ISPRS J. Photogramm. Remote Sens.* **2017**, *132*, 48–60. [CrossRef]

25. Wang, S.; Chen, W.; Xie, S.M.; Azzari, G.; Lobell, D.B. Weakly supervised deep learning for segmentation of remote sensing imagery. *Remote Sens.* **2020**, *12*, 207. [CrossRef]
26. De Castro Filho, H.C.; de Carvalho Júnior, O.A.; de Carvalho, O.L.F.; de Bem, P.P.; dos Santos de Moura, R.; Olino de Albuquerque, A.; Rosa Silva, C.; Guimarães Ferreira, P.H.; Guimarães, R.F.; Gomes, R.A.T. Rice Crop Detection Using LSTM, Bi-LSTM, and Machine Learning Models from Sentinel-1 Time Series. *Remote Sens.* **2020**, *12*, 2655. [CrossRef]
27. Ienco, D.; Interdonato, R.; Gaetano, R.; Ho Tong Minh, D. Combining Sentinel-1 and Sentinel-2 Satellite Image Time Series for land cover mapping via a multi-source deep learning architecture. *ISPRS J. Photogramm. Remote Sens.* **2019**, *158*, 11–22. [CrossRef]
28. Interdonato, R.; Ienco, D.; Gaetano, R.; Ose, K. DuPLO: A DUal view Point deep Learning architecture for time series classification. *ISPRS J. Photogramm. Remote Sens.* **2019**, *149*, 91–104. [CrossRef]
29. Zhong, L.; Hu, L.; Zhou, H. Deep learning based multi-temporal crop classification. *Remote Sens. Environ.* **2019**, *221*, 430–443. [CrossRef]
30. Wieland, M.; Li, Y.; Martinis, S. Multi-sensor cloud and cloud shadow segmentation with a convolutional neural network. *Remote Sens. Environ.* **2019**, *230*, 111203. [CrossRef]
31. Li, Z.; Shen, H.; Cheng, Q.; Liu, Y.; You, S.; He, Z. Deep learning based cloud detection for medium and high resolution remote sensing images of different sensors. *ISPRS J. Photogramm. Remote Sens.* **2019**, *150*, 197–212. [CrossRef]
32. Xie, F.; Shi, M.; Shi, Z.; Yin, J.; Zhao, D. Multilevel cloud detection in remote sensing images based on deep learning. *IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens.* **2017**, *10*, 3631–3640. [CrossRef]
33. Li, Y.; Chen, W.; Zhang, Y.; Tao, C.; Xiao, R.; Tan, Y. Accurate cloud detection in high-resolution remote sensing imagery by weakly supervised deep learning. *Remote Sens. Environ.* **2020**, *250*, 112045. [CrossRef]
34. Li, T.; Shen, H.; Yuan, Q.; Zhang, L. Geographically and temporally weighted neural networks for satellite-based mapping of ground-level PM2.5. *ISPRS J. Photogramm. Remote Sens.* **2020**, *167*, 178–188. [CrossRef]
35. Park, Y.; Kwon, B.; Heo, J.; Hu, X.; Liu, Y.; Moon, T. Estimating PM2.5 concentration of the conterminous United States via interpretable convolutional neural networks. *Environ. Pollut.* **2020**, *256*, 113395. [CrossRef]
36. Shen, H.; Li, T.; Yuan, Q.; Zhang, L. Estimating Regional Ground-Level PM 2.5 Directly From Satellite Top-Of-Atmosphere Reflectance Using Deep Belief Networks. *J. Geophys. Res. Atmos.* **2018**, *123*, 13875–13886. [CrossRef]
37. Wen, C.; Liu, S.; Yao, X.; Peng, L.; Li, X.; Hu, Y.; Chi, T. A novel spatiotemporal convolutional long short-term neural network for air pollution prediction. *Sci. Total Environ.* **2019**, *654*, 1091–1099. [CrossRef]
38. Carranza-García, M.; García-Gutiérrez, J.; Riquelme, J.C. A framework for evaluating land use and land cover classification using convolutional neural networks. *Remote Sens.* **2019**, *11*, 274. [CrossRef]
39. Helber, P.; Bischke, B.; Dengel, A.; Borth, D. Eurosat: A novel dataset and deep learning benchmark for land use and land cover classification. *IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens.* **2019**, *12*, 2217–2226. [CrossRef]
40. Zhang, C.; Sargent, I.; Pan, X.; Li, H.; Gardiner, A.; Hare, J.; Atkinson, P.M. Joint Deep Learning for land cover and land use classification. *Remote Sens. Environ.* **2019**, *221*, 173–187. [CrossRef]
41. Zhang, C.; Harrison, P.A.; Pan, X.; Li, H.; Sargent, I.; Atkinson, P.M. Scale Sequence Joint Deep Learning (SS-JDL) for land use and land cover classification. *Remote Sens. Environ.* **2020**, *237*, 111593. [CrossRef]
42. Huang, B.; Zhao, B.; Song, Y. Urban land-use mapping using a deep convolutional neural network with high spatial resolution multispectral remote sensing imagery. *Remote Sens. Environ.* **2018**, *214*, 73–86. [CrossRef]
43. Huang, F.; Yu, Y.; Feng, T. Automatic extraction of urban impervious surfaces based on deep learning and multi-source remote sensing data. *J. Vis. Commun. Image Represent.* **2019**, *60*, 16–27. [CrossRef]
44. Li, W.; Liu, H.; Wang, Y.; Li, Z.; Jia, Y.; Gui, G. Deep Learning-Based Classification Methods for Remote Sensing Images in Urban Built-Up Areas. *IEEE Access* **2019**, *7*, 36274–36284. [CrossRef]
45. Srivastava, S.; Vargas-Muñoz, J.E.; Tuia, D. Understanding urban landuse from the above and ground perspectives: A deep learning, multimodal solution. *Remote Sens. Environ.* **2019**, *228*, 129–143. [CrossRef]
46. Li, X.; Liu, B.; Zheng, G.; Ren, Y.; Zhang, S.; Liu, Y.; Gao, L.; Liu, Y.; Zhang, B.; Wang, F. Deep learning-based information mining from ocean remote sensing imagery. *Natl. Sci. Rev.* **2020**, nwaa047. [CrossRef]
47. Arellano-Verdejo, J.; Lazcano-Hernandez, H.E.; Cabanillas-Terán, N. ERISNet: Deep neural network for Sargassum detection along the coastline of the Mexican Caribbean. *PeerJ* **2019**, *7*, e6842. [CrossRef]
48. Guo, H.; Wei, G.; An, J. Dark Spot Detection in SAR Images of Oil Spill Using Segnet. *Appl. Sci.* **2018**, *8*, 2670. [CrossRef]
49. Gao, Y.; Gao, F.; Dong, J.; Wang, S. Transferred Deep Learning for Sea Ice Change Detection From Synthetic-Aperture Radar Images. *IEEE Geosci. Remote Sens. Lett.* **2019**, *16*, 1655–1659. [CrossRef]
50. Garcia-Garcia, A.; Orts-Escolano, S.; Oprea, S.; Villena-Martinez, V.; Garcia-Rodriguez, J. A Review on Deep Learning Techniques Applied to Semantic Segmentation. *arXiv* **2017**, arXiv:1704.06857.
51. Guo, Y.; Liu, Y.; Georgiou, T.; Lew, M.S. A review of semantic segmentation using deep neural networks. *Int. J. Multimed. Inf. Retr.* **2018**, *7*, 87–93. [CrossRef]
52. Garcia-Garcia, A.; Orts-Escolano, S.; Oprea, S.; Villena-Martinez, V.; Martinez-Gonzalez, P.; Garcia-Rodriguez, J. A survey on deep learning techniques for image and video semantic segmentation. *Appl. Soft Comput.* **2018**, *70*, 41–65. [CrossRef]
53. Geng, Q.; Zhou, Z.; Cao, X. Survey of recent progress in semantic image segmentation with CNNs. *Sci. China Inf. Sci.* **2018**, *61*, 1–18. [CrossRef]

54. Lateef, F.; Ruichek, Y. Survey on semantic segmentation using deep learning techniques. *Neurocomputing* **2019**, *338*, 321–348. [CrossRef]
55. Yu, H.; Yang, Z.; Tan, L.; Wang, Y.; Sun, W.; Sun, M.; Tang, Y. Methods and datasets on semantic segmentation: A review. *Neurocomputing* **2018**, *304*, 82–103. [CrossRef]
56. He, K.; Gkioxari, G.; Dollar, P.; Girshick, R. Mask R-CNN. *IEEE Trans. Pattern Anal. Mach. Intell.* **2020**, *42*, 386–397. [CrossRef]
57. Dai, J.; He, K.; Sun, J. Instance-Aware Semantic Segmentation via Multi-task Network Cascades. In Proceedings of the 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, USA, 26 June–1 July 2016; pp. 3150–3158.
58. Pinheiro, P.O.; Collobert, R.; Dollar, P. Learning to Segment Object Candidates. In Proceedings of the 28th International Conference on Neural Information Processing Systems (NIPS'15), Montreal, QC, Canada, 7–12 December 2015; pp. 1990–1998.
59. Pinheiro, P.O.; Lin, T.Y.; Collobert, R.; Dollár, P. Learning to refine object segments. In Proceedings of the 14th European Conference on Computer Vision (ECCV 2016), Amsterdam, The Netherlands, 11–14 October 2016; Volume 9905, pp. 75–91. [CrossRef]
60. Arnab, A.; Torr, P.H.S. Pixelwise Instance Segmentation with a Dynamically Instantiated Network. In Proceedings of the 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Honolulu, HI, USA, 21–26 July 2017; pp. 879–888.
61. Bai, M.; Urtasun, R. Deep Watershed Transform for Instance Segmentation. In Proceedings of the 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Honolulu, HI, USA, 21–26 July 2017; pp. 2858–2866.
62. Kirillov, A.; Levinkov, E.; Andres, B.; Savchynskyy, B.; Rother, C. InstanceCut: From Edges to Instances with MultiCut. In Proceedings of the 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Honolulu, HI, USA, 21–26 July 2017; pp. 7322–7331.
63. Liu, S.; Qi, L.; Qin, H.; Shi, J.; Jia, J. Path Aggregation Network for Instance Segmentation. In Proceedings of the 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–23 June 2018; pp. 8759–8768.
64. Li, Y.; Qi, H.; Dai, J.; Ji, X.; Wei, Y. Fully Convolutional Instance-Aware Semantic Segmentation. In Proceedings of the 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Honolulu, HI, USA, 21–26 July 2017; pp. 4438–4446.
65. Cai, Z.; Vasconcelos, N. Cascade R-CNN: Delving Into High Quality Object Detection. In Proceedings of the 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–23 June 2018; pp. 6154–6162.
66. Chen, K.; Ouyang, W.; Loy, C.C.; Lin, D.; Pang, J.; Wang, J.; Xiong, Y.; Li, X.; Sun, S.; Feng, W.; et al. Hybrid Task Cascade for Instance Segmentation. In Proceedings of the 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Long Beach, CA, USA, 16–21 June 2019; pp. 4969–4978.
67. Huang, Z.; Huang, L.; Gong, Y.; Huang, C.; Wang, X. Mask Scoring R-CNN. In Proceedings of the 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Long Beach, CA, USA, 16–21 June 2019; pp. 6402–6411.
68. Su, H.; Wei, S.; Liu, S.; Liang, J.; Wang, C.; Shi, J.; Zhang, X. HQ-ISNet: High-Quality Instance Segmentation for Remote Sensing Imagery. *Remote Sens.* **2020**, *12*, 989. [CrossRef]
69. Asgari Taghanaki, S.; Abhishek, K.; Cohen, J.P.; Cohen-Adad, J.; Hamarneh, G. Deep semantic segmentation of natural and medical images: A review. *Artif. Intell. Rev.* **2020**. [CrossRef]
70. Deng, S.; Zhang, X.; Yan, W.; Chang, E.I.C.; Fan, Y.; Lai, M.; Xu, Y. Deep learning in digital pathology image analysis: A survey. *Front. Med.* **2020**. [CrossRef]
71. Jiang, Y.; Li, C. Convolutional Neural Networks for Image-Based High-Throughput Plant Phenotyping: A Review. *Plant. Phenomics* **2020**, *2020*, 4152816. [CrossRef]
72. Ruiz-Santaquiteria, J.; Bueno, G.; Deniz, O.; Vallez, N.; Cristobal, G. Semantic versus instance segmentation in microscopic algae detection. *Eng. Appl. Artif. Intell.* **2020**, *87*, 103271. [CrossRef]
73. Qiao, Y.; Truman, M.; Sukkarieh, S. Cattle segmentation and contour extraction based on Mask R-CNN for precision livestock farming. *Comput. Electron. Agric.* **2019**, *165*, 104958. [CrossRef]
74. Xu, B.; Wang, W.; Falzon, G.; Kwan, P.; Guo, L.; Chen, G.; Tait, A.; Schneider, D. Automated cattle counting using Mask R-CNN in quadcopter vision system. *Comput. Electron. Agric.* **2020**, *171*, 105300. [CrossRef]
75. Champ, J.; Mora-Fallas, A.; Goëau, H.; Mata-Montero, E.; Bonnet, P.; Joly, A. Instance segmentation for the fine detection of crop and weed plants by precision agricultural robots. *Appl. Plant. Sci.* **2020**, *8*, 1–10. [CrossRef] [PubMed]
76. Zhang, Q.; Liu, Y.; Gong, C.; Chen, Y.; Yu, H. Applications of Deep Learning for Dense Scenes Analysis in Agriculture: A Review. *Sensors* **2020**, *20*, 1520. [CrossRef]
77. Yekeen, S.T.; Balogun, A.; Wan Yusof, K.B. A novel deep learning instance segmentation model for automated marine oil spill detection. *ISPRS J. Photogramm. Remote Sens.* **2020**, *167*, 190–200. [CrossRef]
78. Li, Q.; Mou, L.; Hua, Y.; Sun, Y.; Jin, P.; Shi, Y.; Zhu, X.X. Instance segmentation of buildings using keypoints. *arXiv* **2020**, arXiv:2006.03858.
79. Wen, Q.; Jiang, K.; Wang, W.; Liu, Q.; Guo, Q.; Li, L.; Wang, P. Automatic Building Extraction from Google Earth Images under Complex Backgrounds Based on Deep Instance Segmentation Network. *Sensors* **2019**, *19*, 333. [CrossRef]
80. Mou, L.; Zhu, X.X. Vehicle Instance Segmentation from Aerial Image and Video Using a Multitask Learning Residual Fully Convolutional Network. *IEEE Trans. Geosci. Remote Sens.* **2018**, *56*, 6699–6711. [CrossRef]
81. Feng, Y.; Diao, W.; Zhang, Y.; Li, H.; Chang, Z.; Yan, M.; Sun, X.; Gao, X. Ship Instance Segmentation from Remote Sensing Images Using Sequence Local Context Module. In Proceedings of the 2019 IEEE International Geoscience and Remote Sensing Symposium (IGARSS 2019), Yokohama, Japan, 28 July–2 August 2019; pp. 1025–1028.

82. Yu, Y.; Zhang, K.; Yang, L.; Zhang, D. Fruit detection for strawberry harvesting robot in non-structural environment based on Mask-RCNN. *Comput. Electron. Agric.* **2019**, *163*, 104846. [CrossRef]
83. Wei, X.S.; Xie, C.W.; Wu, J.; Shen, C. Mask-CNN: Localizing parts and selecting descriptors for fine-grained bird species categorization. *Pattern Recognit.* **2018**, *76*, 704–714. [CrossRef]
84. Audebert, N.; Le Saux, B.; Lefèvre, S. Beyond RGB: Very high resolution urban remote sensing with multimodal deep networks. *ISPRS J. Photogramm. Remote Sens.* **2018**, *140*, 20–32. [CrossRef]
85. De Albuquerque, A.O.; de Carvalho Júnior, O.A.; de Carvalho, O.L.F.; de Bem, P.P.; Ferreira, P.H.G.; dos Santos de Moura, R.; Silva, C.R.; Trancoso Gomes, R.A.; Fontes Guimarães, R. Deep Semantic Segmentation of Center Pivot Irrigation Systems from Remotely Sensed Data. *Remote Sens.* **2020**, *12*, 2159. [CrossRef]
86. Martins, V.S.; Kaleita, A.L.; Gelder, B.K.; da Silveira, H.L.F.; Abe, C.A. Exploring multiscale object-based convolutional neural network (multi-OCNN) for remote sensing image classification at high spatial resolution. *ISPRS J. Photogramm. Remote Sens.* **2020**, *168*, 56–73. [CrossRef]
87. Yi, Y.; Zhang, Z.; Zhang, W.; Zhang, C.; Li, W.; Zhao, T. Semantic Segmentation of Urban Buildings from VHR Remote Sensing Imagery Using a Deep Convolutional Neural Network. *Remote Sens.* **2019**, *11*, 1774. [CrossRef]
88. Audebert, N.; Le Saux, B.; Lefèvre, S. Segment-before-Detect: Vehicle Detection and Classification through Semantic Segmentation of Aerial Images. *Remote Sens.* **2017**, *9*, 368. [CrossRef]
89. Mahdianpari, M.; Salehi, B.; Rezaee, M.; Mohammadimanesh, F.; Zhang, Y. Very Deep Convolutional Neural Networks for Complex Land Cover Mapping Using Multispectral Remote Sensing Imagery. *Remote Sens.* **2018**, *10*, 1119. [CrossRef]
90. Russell, B.C.; Torralba, A.; Murphy, K.P.; Freeman, W.T. LabelMe: A Database and Web-Based Tool for Image Annotation. *Int. J. Comput. Vis.* **2008**, *77*, 157–173. [CrossRef]
91. Deng, J.; Dong, W.; Socher, R.; Li, L.-J.; Kai, L.; Fei-Fei, L. ImageNet: A large-scale hierarchical image database. In Proceedings of the 2009 IEEE Conference on Computer Vision and Pattern Recognition, Miami, FL, USA, 22–24 June 2009; pp. 248–255.
92. Everingham, M.; Eslami, S.M.A.; Van Gool, L.; Williams, C.K.I.; Winn, J.; Zisserman, A. The Pascal Visual Object Classes Challenge: A Retrospective. *Int. J. Comput. Vis.* **2015**, *111*, 98–136. [CrossRef]
93. Cordts, M.; Omran, M.; Ramos, S.; Rehfeld, T.; Enzweiler, M.; Benenson, R.; Franke, U.; Roth, S.; Schiele, B. The Cityscapes Dataset for Semantic Urban Scene Understanding. In Proceedings of the 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, USA, 26 June–1 July 2016; Volume 29, pp. 3213–3223.
94. Kuznetsova, A.; Rom, H.; Alldrin, N.; Uijlings, J.; Krasin, I.; Pont-Tuset, J.; Kamali, S.; Popov, S.; Malloci, M.; Kolesnikov, A.; et al. The Open Images Dataset V4. *Int. J. Comput. Vis.* **2020**, *128*, 1956–1981. [CrossRef]
95. Lin, T.-Y.; Maire, M.; Belongie, S.; Hays, J.; Perona, P.; Ramanan, D.; Dollár, P.; Zitnick, C.L. Microsoft COCO: Common Objects in Context. In Proceedings of the 13th European Conference on Computer Vision (ECCV 2014), Zurich, Switzerland, 6–12 September 2014; Volume 8693, pp. 740–755, ISBN 978-3-319-10601-4.
96. Wu, Y.; Kirillov, A.; Massa, F.; Lo, W.-Y.; Girshick, R. Detectron2. 2019. Available online: lhttps://github.com/facebookresearch/detectron2 (accessed on 14 November 2020).
97. Rundquist, D.C.; Hoffman, R.O.; Carlson, M.P.; Cook, A.E. Nebraska center-pivot inventory: An example of operational satellite remote sensing on a long-term basis. *Photogramm. Eng. Remote Sensing* **1989**, *55*, 587–590.
98. Heller, R.C.; Johnson, K.A. Estimating irrigated land acreage from Landsat imagery. *Photogramm. Eng. Remote Sensing* **1979**, *45*, 1379–1386.
99. Agência Nacional de Águas. *Levantamento da Agricultura Irrigada por Pivôs Centrais no Brasil (1985–2017)*; ANA: Brasilia, Brazil, 2019.
100. Agência Nacional de Águas. *Levantamento da Agricultura Irrigada por Pivôs Centrais no Brasil—2014: Relatório Síntese*; ANA: Brasilia, Brazil, 2016; ISBN 9788582100349.
101. Ferreira, E.; De Toledo, J.H.; Dantas, A.A.A.; Pereira, R.M. Cadastral maps of irrigated areas by center pivots in the State of Minas Gerais, using CBERS-2B/CCD satellite imaging. *Eng. Agric.* **2011**, *31*, 771–780. [CrossRef]
102. Martins, J.D.; Bohrz, I.S.; Fredrich, M.; Veronez, R.P.; Kunz, G.A.; Tura, E.F. Levantamento da área irrigada por pivô central no Estado do Rio Grande do Sul. *Irrig. Botucatu* **2016**, *21*, 300–311. [CrossRef]
103. Zhang, C.; Yue, P.; Di, L.; Wu, Z. Automatic Identification of Center Pivot Irrigation Systems from Landsat Images Using Convolutional Neural Networks. *Agriculture* **2018**, *8*, 147. [CrossRef]
104. Saraiva, M.; Protas, É.; Salgado, M.; Souza, C. Automatic Mapping of Center Pivot Irrigation Systems from Satellite Images Using Deep Learning. *Remote Sens.* **2020**, *12*, 558. [CrossRef]
105. Shermeyer, J.; Hossler, T.; van Etten, A.; Hogan, D.; Lewis, R.; Kim, D. RarePlanes: Synthetic Data Takes Flight. *arXiv* **2020**, arXiv:2006.02963.
106. Xia, G.; Bai, X.; Ding, J.; Zhu, Z.; Belongie, S.; Luo, J.; Datcu, M.; Pelillo, M.; Zhang, L. DOTA: A Large-Scale Dataset for Object Detection in Aerial Images. In Proceedings of the 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–23 June 2018; pp. 3974–3983.
107. Zamir, S.W.; Arora, A.; Gupta, A.; Khan, S.; Sun, G.; Khan, F.S.; Zhu, F.; Shao, L.; Xia, G.S.; Bai, X. iSAID: A large-scale dataset for instance segmentation in aerial images. *arXiv* **2019**, arXiv:1905.12886.
108. Van Etten, A.; Lindenbaum, D.; Bacastow, T. SpaceNet: A remote sensing dataset and challenge series. *arXiv* **2018**, arXiv:1807.01232.

109. Althoff, D.; Rodrigues, L.N. The expansion of center-pivot irrigation in the cerrado biome. *Irriga* **2019**, *1*, 56–61. [CrossRef]
110. Brunckhorst, A.; de Souza Bias, E. Aplicação de sig na gestão de conflitos pelo uso da água na porção goiana da bacia hidrográfica do rio são Marcos, município de Cristalina—GO. *Geociencias* **2014**, *33*, 23–31.
111. Silva, L.M.D.C.; Da Hora, M.D.A.G.M. Conflito pelo Uso da Água na Bacia Hidrográfica do Rio São Marcos: O Estudo de Caso da UHE Batalha. *Engevista* **2014**, *17*, 166. [CrossRef]
112. De Oliveira, S.N.; de Carvalho Júnior, O.A.; Gomes, R.A.T.; Guimarães, R.F.; McManus, C.M. Landscape-fragmentation change due to recent agricultural expansion in the Brazilian Savanna, Western Bahia, Brazil. *Reg. Environ. Chang.* **2017**, *17*, 411–423. [CrossRef]
113. De Oliveira, S.N.; de Carvalho Júnior, O.A.; Trancoso Gomes, R.A.; Fontes Guimarães, R.; McManus, C.M. Deforestation analysis in protected areas and scenario simulation for structural corridors in the agricultural frontier of Western Bahia, Brazil. *Land Use Policy* **2017**, *61*, 40–52. [CrossRef]
114. Pousa, R.; Costa, M.H.; Pimenta, F.M.; Fontes, V.C.; Castro, M. Climate change and intense irrigation growth in Western Bahia, Brazil: The urgent need for hydroclimatic monitoring. *Water* **2019**, *11*, 933. [CrossRef]
115. Kelly, A. Cocosynth. Available online: https://github.com/akTwelve/cocosynth (accessed on 30 August 2020).
116. He, K.; Gkioxari, G.; Dollar, P.; Girshick, R. Mask R-CNN. In Proceedings of the 2017 IEEE International Conference on Computer Vision (ICCV), Venice, Italy, 22–29 October 2017; pp. 2980–2988.
117. Girshick, R.; Donahue, J.; Darrell, T.; Malik, J. Rich Feature Hierarchies for Accurate Object Detection and Semantic Segmentation. In Proceedings of the 2014 IEEE Conference on Computer Vision and Pattern Recognition, Columbus, OH, USA, 23–28 June 2014; Volume 1, pp. 580–587.
118. Girshick, R. Fast R-CNN. In Proceedings of the 2015 IEEE International Conference on Computer Vision (ICCV), Santiago, Chile, 7–13 December 2015; pp. 1440–1448.
119. Ren, S.; He, K.; Girshick, R.; Sun, J. Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks. *IEEE Trans. Pattern Anal. Mach. Intell.* **2017**, *39*, 1137–1149. [CrossRef]
120. He, K.; Zhang, X.; Ren, S.; Sun, J. Deep Residual Learning for Image Recognition. In Proceedings of the 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, USA, 26 June–1 July 2016; Volume 45, pp. 770–778.
121. Xie, S.; Girshick, R.; Dollar, P.; Tu, Z.; He, K. Aggregated Residual Transformations for Deep Neural Networks. In Proceedings of the 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Honolulu, HI, USA, 21–26 July 2017; pp. 5987–5995.
122. Lin, T.-Y.; Dollar, P.; Girshick, R.; He, K.; Hariharan, B.; Belongie, S. Feature Pyramid Networks for Object Detection. In Proceedings of the 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Honolulu, HI, USA, 21–26 July 2017; pp. 936–944.
123. Girshick, R.; Donahue, J.; Darrell, T.; Malik, J. Region-Based Convolutional Networks for Accurate Object Detection and Segmentation. *IEEE Trans. Pattern Anal. Mach. Intell.* **2016**, *38*, 142–158. [CrossRef]
124. Dai, Z.; Heckel, R. Channel Normalization in Convolutional Neural Network avoids Vanishing Gradients. *arXiv* **2019**, arXiv:1907.09539.
125. Bolya, D.; Zhou, C.; Xiao, F.; Lee, Y.J. YOLACT: Real-Time Instance Segmentation. In Proceedings of the 2019 IEEE/CVF International Conference on Computer Vision (ICCV), Seoul, Korea, 27 October–2 November 2019; pp. 9156–9165.
126. Bolya, D.; Zhou, C.; Xiao, F.; Lee, Y.J. YOLACT++: Better Real-time Instance Segmentation. *IEEE Trans. Pattern Anal. Mach. Intell.* **2020**, 1. [CrossRef]
127. Zhao, W.; Du, S.; Emery, W.J. Object-Based Convolutional Neural Network for High-Resolution Imagery Classification. *IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens.* **2017**, *10*, 3386–3396. [CrossRef]
128. Zhang, C.; Sargent, I.; Pan, X.; Li, H.; Gardiner, A.; Hare, J.; Atkinson, P.M. An object-based convolutional neural network (OCNN) for urban land use classification. *Remote Sens. Environ.* **2018**, *216*, 57–70. [CrossRef]
129. Deng, Z.; Sun, H.; Zhou, S.; Zhao, J.; Lei, L.; Zou, H. Multi-scale object detection in remote sensing imagery with convolutional neural networks. *ISPRS J. Photogramm. Remote Sens.* **2018**, *145*, 3–22. [CrossRef]
130. Su, H.; Wei, S.; Yan, M.; Wang, C.; Shi, J.; Zhang, X. Object Detection and Instance Segmentation in Remote Sensing Imagery Based on Precise Mask R-CNN. In Proceedings of the 2019 IEEE International Geoscience and Remote Sensing Symposium (IGARSS 2019), Yokohama, Japan, 28 July–2 August 2019; pp. 1454–1457.
131. Pang, J.; Li, C.; Shi, J.; Xu, Z.; Feng, H. R2-CNN: Fast Tiny Object Detection in Large-Scale Remote Sensing Images. *IEEE Trans. Geosci. Remote Sens.* **2019**, *57*, 5512–5524. [CrossRef]
132. Li, K.; Wan, G.; Cheng, G.; Meng, L.; Han, J. Object detection in optical remote sensing images: A survey and a new benchmark. *ISPRS J. Photogramm. Remote Sens.* **2020**, *159*, 296–307. [CrossRef]
133. Zhao, K.; Kang, J.; Jung, J.; Sohn, G. Building Extraction from Satellite Images Using Mask R-CNN with Building Boundary Regularization. In Proceedings of the 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–23 June 2018; pp. 242–246.