# Bounding Box-Free Instance Segmentation Using Semi-Supervised Iterative Learning for Vehicle Detection

Osmar Luiz Ferreira de Carvalho, *Member, IEEE*, Osmar Abílio de Carvalho Júnior, *Member, IEEE*, Anesmar Olino de Albuquerque, *Student Member, IEEE*, Nickolas Castro Santana, *Member, IEEE*, Renato Fontes Guimarães, *Member, IEEE*, Roberto Arnaldo Trancoso Gomes, *Member, IEEE*, and Díbio Leandro Borges, *Senior Member, IEEE*

***Abstract*—Vehicle classification is a hot computer vision topic, with studies ranging from ground-view to top-view imagery. Top-view images allow understanding city patterns, traffic management, among others. However, there are some difficulties for pixel-wise classification: most vehicle classification studies use object detection methods, and most publicly available datasets are designed for this task, creating instance segmentation datasets is laborious, and traditional instance segmentation methods underperform on this task since the objects are small. Thus, the present research objectives are as follows: first, propose a novel semisupervised iterative learning approach using the geographic information system software, second, propose a box-free instance segmentation approach, and third, provide a city-scale vehicle dataset. The iterative learning procedure considered the following: first, labeling a few vehicles from the entire scene, second, choosing training samples near those areas, third, training the deep learning model (U-net with efficient-net-B7 backbone), fourth, classifying the whole scene, fifth, converting the predictions into shapefile, sixth, correcting areas with wrong predictions, seventh, including them in the training data, eighth repeating until results are satisfactory. We considered vehicle interior and borders to separate instances using a semantic segmentation model. When removing the borders, the vehicle interior becomes isolated, allowing for unique object identification. Our procedure is very efficient and accurate for generating data iteratively, which resulted in 122 567 mapped vehicles. Metrics-wise, our method presented higher intersection over union when compared to box-based methods (82% against 72%), and per-object metrics surpassed 90% for precision and recall.**

***Index Terms*—Aerial image, anchor-free, deep learning (DL), instance segmentation.**

Manuscript received January 9, 2022; revised March 16, 2022; accepted April 17, 2022. Date of publication April 21, 2022; date of current version May 11, 2022. *(Corresponding author: Osmar Abílio de Carvalho Júnior.)*

Osmar Luiz Ferreira de Carvalho is with the Department of Computer Science, University of Brasília, Brasília 30332, Brazil, and also with the Department of Geography, University of Brasília, Brasília 70910-900, Brazil (e-mail: osmarcarvalho@ieee.org).

Osmar Abílio de Carvalho Júnior, Anesmar Olino de Albuquerque, Nickolas Castro Santana, Renato Fontes Guimarães, and Roberto Arnaldo Trancoso Gomes are with the Department of Geography, University of Brasília, Brasília 70910-900, Brazil (e-mail: osmarjr@unb.br; anesmar@ieee.org; nickolas.santana@unb.br; renatofg@unb.br; robertogomes@unb.br).

Díbio Leandro Borges is with the Department of Computer Science, University of Brasília, Brasília 30332, Brazil (e-mail: dibio@unb.br).

Digital Object Identifier 10.1109/JSTARS.2022.3169128

## I. Introduction

USUALLY, the city's infrastructure was not designed to absorb population growth and road traffic, which has reached high congestion levels in many urban centers worldwide. The accentuated growth in the number of vehicles makes monitoring and managing urban traffic highly complex and necessary. In this context, automatic vehicle detection based on remote sensing images is a powerful tool for various applications, such as traffic monitoring, air pollution, congestion studies, public safety, parking utilization, disaster management, and rescue missions. Periodic image acquisition provides information on the number and location of vehicles in different urban environments, allowing coverage of large areas and proper monitoring of moving targets.

Vehicle detection is a widely studied topic in the computer vision community, containing several studies with ground-view and aerial-view images. These two approaches present marked differences in vehicle representation, in which ground images emphasize the vehicle faces, while the top view of the vehicle acquires straight shapes [1], [2]. Another significant difference is that the vehicle's spatial resolution in aerial images is significantly lower than in terrestrial images. In-ground view images, several literature reviews address advanced driver assistance systems for autonomous vehicles using image processing and vehicle detection from various onboard handling sensors, such as radar, monocular camera, and camera binocular [3]–[5]. In addition, several studies use images from surveillance cameras on roads [6], on top of buildings [7], pedestrian bridges [8], among others.

Despite the broad applicability of ground images and videos, vehicle detection from high-resolution aerial and satellite images allows for a synoptic understanding of city patterns, guiding crucial public policies, such as urban planning and traffic management. Vehicle detection using aerial view imagery includes different strategies and sensors, such as unmanned aerial vehicles (UAV), airplanes, or orbital platforms, which provide data at different heights and resolutions.

Even though skilled professionals may easily distinguish vehicles from different urban features, the rapid and automatic classification is a challenging task since the vehicles:

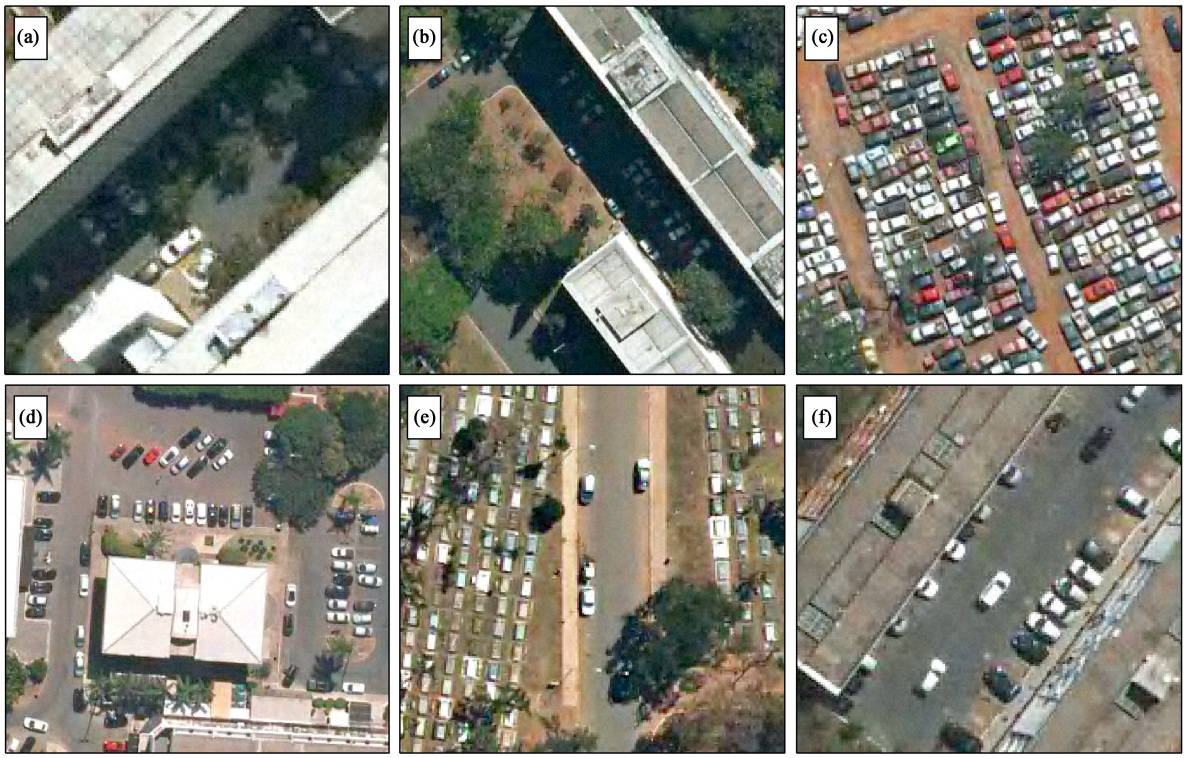


Fig. 1. Six examples ((a), (b), (c), (d), (e), and (f)) of difficult regions to classify cars in the urban setting.

1) are small objects;
2) present high variability in shape, color, and size;
3) appear in different background settings;
4) present different brightness and contrasts among the city;
5) may be crowded (e.g., parking lots);
6) may be occluded by other objects, such as trees and buildings; and
7) have many look-alikes in the city.

Fig. 1 shows six examples of difficult areas to identify the vehicles, where (a) and (b) present shadows, (c) and (d) show a large concentration of vehicles, (e) presents look-alikes (the tombs are very similar to cars when seen from this angle), and (f) presents occluded cars by the building roof.

Currently, the deep learning (DL) methods represent state-of-the-art vehicle detection, surpassing traditional algorithms. These advances are strongly related to convolutional neural networks (CNN), which apply kernels along with the image, obtaining low, middle, and high-level features, enhancing the classification results. Vehicle detection using DL may present different approaches, such as object detection [9], semantic segmentation [10], and instance segmentation [11]. In object detection, the DL outputs bounding boxes around the car. Instance segmentation generates bounding boxes and a segmentation mask, and semantic segmentation outputs a class-aware segmentation mask.

Most studies on vehicles address object detection that focuses on the delineation of the targets' bounding box, while instance segmentation, which aims at mapping each object at the pixel level, is still little explored. A challenge in the individual segmentation of vehicles is the lower performance for small objects that, when they are very close, coalesce into a single group [12], [13]. Furthermore, deep instance segmentation methods require a large amount of data, especially considering small object detection. Therefore, training requires a much more complex annotation (since it requires the polygons from each object), containing all possible variations and apparition locations to not depend on a given scenario. The common objects in context (COCO) [14] dataset defined small objects with less than $32^2$ pixels and results considering the small objects are nearly half of the performance on medium and large objects.

More recently, artificial intelligence has an upcoming trend that aims to enhance results and practical solutions by using a data-centric rather than a model-centric approach. The central concept behind this is that the model performance is already very high and that enhancing the data would bring better benefits. One pillar of the model-centric approach is the selection of more informative samples within the dataset. In this context, active learning is a promising methodology to obtain quality labeled data sequentially. In remote sensing, images often present vast dimensions, and the integration of commonly used GIS software may be an excellent ally for active learning in object detection, since:

1) we may see the entire data at once;
2) it is very straightforward to manipulate and correct polygon data;
3) we may use other facilities, such as polygon shapefiles to choose where to gather the data.

The present research aims to advance in three fields (data generation through iterative learning, DL method, and dataset).

1) *Iterative learning procedure for data generation*: A novel proposition for integration of DL with commonly used GIS software by iteratively correcting erroneous areas, being less time-consuming and laborious.
2) *Bounding box-free instance segmentation*: A novel instance segmentation method that uses object interiors and contours to isolate them and output separate instances.
3) *BSB vehicle dataset*: A city-scale dataset with polygons shapefiles.

## II. Related Works

Different strategies have been developed and described for vehicle detection through aerial and orbital images in the last two decades. In this trajectory, the following two main approaches stand out [15]–[17]: a) methods based on superficial learning and b) DL-based methods.

### A. Early Vehicle Detection Studies Using a Shallow-Learning-Based Approach

Considering vehicle detection approaches based on superficial learning, Hinz [18] proposed a generic subdivision into explicit and implicit models. The explicit model describes a vehicle in 2-D or 3-D (representation of a box or wire-frame structure), considering the car detection from a "top-down" or "bottom-up" model. The implicit model considers the collection of multiple features of a region of the image and their statistics gathered in vectors followed by a classification process (single classifier, combination of classifiers, or hierarchical model). In the present analysis, we considered the following groups of algorithms: 1) pixel-wise classification and segmentation (including threshold segmentation method, segmentation based on pixel clustering, segmentation based on edge detection and region growth method, segmentation based on inter-frame difference or background difference); 2) object-based classification; object detection (obtaining the bounding box without vehicle segmentation) from multiple features and machine learning within a sliding window approach.

The threshold segmentation method was widely used in different pre-processed images to highlight vehicles, such as principal component analysis, Bayesian background transformation, and gradient-based method [19]; Morphological grayscale method and background difference (vehicle enhancement by subtraction between the original image and the road background image) [20]. Cheng *et al.* [21] perform pixel-wise classification for vehicle detection using dynamic Bayesian networks (DBNs), considering features that comprise pixel-level information and the relationship between neighboring pixels in a region (location analysis of features and color attributes).

Object-based methods use image segmentation to split an image into separated regions and classify them instead of pixels [22]. Different vehicle detection surveys use object-oriented image classification, considering the following:

1) eCognition classification [23];
2) segmentation using Otsu Threshold, feature extraction (geometric-shape properties, gray level features, and Hu moments), and statistical classifier [24];
3) superpixel-based image segmentation, HOG features, and support vector machines (SVM) [25].

Vehicle detection methods have increased significantly by combining more robust descriptor extraction procedures with machine learning methods for object detection (see Table I). Therefore, vehicle detection uses an image scan through a pre-trained classifier. Among the methods of extraction and selection of features, the most used were: Haar-like features, histogram of oriented gradient (HoG), histogram of Gabor coefficient (HGC), and local binary patterns (LBP), local steering kernel (LSK), bag-of-words (BoW), and scale invariant feature transform (SIFT). Several studies have improved the description of cars by combining different resource extraction methods. The most used machine learning methods were the SVM and Adaptive Boosting (AdaBoost) in the classification step. However, the literature also describes the use of other methods to compare and improve detection accuracy and efficiency, such as k-nearest neighbor (k-NN), decision trees (DT), random forests (RF), DBN, partial least squares (PLS). Some associations between feature extraction methods and classifiers had more significant propagation for detecting vehicles, such as HoG + SVM [26] and Haar-like + AdaBoost called Viola–Jones [27]. However, the shallow-learning-based methods do not sufficiently describe and generalize vehicle detection in complex backgrounds. Some studies to minimize errors have restricted vehicle detection to certain circumstances: 1) only along roads, considering the use of masks from a buffer area [20], [28]–[31]; 2) exclusion of objects elevated above a certain height from the DEM (e.g., buildings and vegetation) [32]; and correlation of cars in consecutive frames [33]. Also, most of these methods are sensitive to the in-plane rotation of objects (detecting only in a specific orientation) and to changes in lighting such as Viola–Jones.

In the transition from traditional to DL methods, some studies use deep architecture only to extract highly descriptive features combined with a machine learning classifier. In this approach, the following propositions stand out: deep Boltzmann machines (DBMs) and weakly supervised learning [34], multilayer deep resource generation model using DBMs and multiscale hough forest model [35], [36], CNN and Exemplar-SVMs [37], and CNN and SVM [38].

### B. DL-Based Vehicle Detection

A significant milestone in CNN's dominance in computer vision was its success in the ImageNet large scale visual recognition challenge in 2012 [53]. DL-based vehicle detection studies have intensified in the following years, with an annual increase making it the dominant method today. DL architecture networks perform better than shallow learning-based methods due to the following reasons [54]:

1) operates both for feature extraction and classification;
2) CNN improves automatic feature generation with the ability to learn local characteristics of different orders, inherently exploiting spatial dependence;
3) less time-consuming.

TABLE I
STUDIES DEVELOPED FOR THE DETECTION OF CARS USING DIFFERENT FEATURE EXTRACTION APPROACHES (SHALLOW-LEARNING-BASED FEATURES) AND CLASSIFICATION, IN WHICH THE FEATURE EXTRACTION METHODS DESCRIBED ARE: COLOR PROBABILITY MAPS (CPM), HAAR-LIKE FEATURES (HLF), HISTOGRAM OF GABOR COEFFICIENTS (HGC), HISTOGRAM OF ORIENTED GRADIENTS (HOG), LOCAL BINARY PATTERNS (LBP), LOCAL STEERING KERNEL (LSK), LOCAL TERNARY PATTERN (LTP), OPPONENT HISTOGRAM (OH), SCALE INVARIANT FEATURE TRANSFORM (SIFT), AND INTEGRAL CHANNEL FEATURES (ICFS)

| Article | Features | Classifier | Image |
|---|---|---|---|
| [39] | HoG, Hlf, LBP | AdaBoost | aerial |
| [31] | HoG, Hlf, LBP | AdaBoost | aerial |
| [40] | LBP, HoG, and Hlf | AdaBoost | aerial |
| [33] | HoG | SVM | UAV |
| [41] | HoG and HGH | k-NN, SVM, DT, and RF | aerial |
| [42] | HoG, CPM, and pairs of pixel comparisons | PLS | aerial |
| [43] | HoG and Hlf | AdaBoost and SVM | WAMI |
| [44] | HoG, LBP, and OH | SVM | aerial |
| [32] | HoG | AdaBoost | aerial |
| [28] | SIFT | SVM | UAV |
| [29] | HoG | SVM | UAV |
| [30] | Hlf | AdaBoost and SVM | aerial |
| [45] | ICFs + HoG | AdaBoost | UAV and GE |
| [46] | HoG | SVM and Causal MRF | UAV |
| [47] | HOG, LBP, and LTP | SVM, DPM, template matching, and Hough Forest | aerial |
| [48] | HoG and Hlf | SVM and AdaBoost | UAV |
| [49] | SIFT | Multi-Instance Learning | satellite |
| [50] | Hlf + Road Orientation Adjustment | AdaBoost | UAV |
| [51] | LSK + bag-of-words (BoW) | SVM | UAV and satellite |
| [52] | LSK + vector of locally aggregated descriptors (VLAD) | Directed-Acyclic-Graph SVM | aerial |

The classification methods are AdaBoost, DT, deformable part model (DPM), DBN, k-NN, PLS, RF, and SVM. The images used in this article are UAV, google earth (GE), and wide area motion imagery (WAMI).

Different DL approaches have been applied in vehicle detection, such as object detection, semantic segmentation, and instance segmentation.

*1) Object Detection:* Vehicle studies using object detection are dominant due to fast target detection, improving real-time monitoring efficiency. However, these methods do not allow a precise mapping of their contours obtained with semantic and instance segmentation. Table II presents the main studies of vehicles using object detection methods. A subdivision of the object detection algorithms is two-stage object detection and one-stage object detection.

Two-step methods first generate several bounding boxes around potential objects called region proposals, and then a classifier determines the object's presence. The classification for each potential object slows down the process, focusing on detection accuracy. As examples of two-stage object detection algorithms highlight regions with CNN features (R-CNN) [55], its variants fast R-CNN [56], faster R-CNN [57], and mask R-CNN [58].

One-stage object detection processes images through a single neural network, detecting, and classifying multiple objects simultaneously and ensuring speed. These methods focus on the detection speed but have limitations to detecting crowded groups of small objects. Among these algorithms, you only look once (YOLO) [59], you only look twice (YOLT) [60], and single-shot multibox detector (SSD) [61] are the most prevalent.

*2) Semantic and Instance Segmentation:* Vehicle studies with semantic and instance segmentation present less quantity than those developed with object detection methods. Tayara *et al.* [13] performed a fully convolutional regression network, whose training stage uses the input image and ground truth data that describes each vehicle as a 2-D Gaussian function distribution. Therefore, the vehicle's original format acquires a simplified elliptical shape in the ground truth and output images. The vehicle segmentation uses a threshold value in the predicted density map, generating a binary mask. Although the method avoids grouping cars and favors counting, vehicles take on a different form described by the Gaussian function, which has a low precision at the pixel level. In contrast, Mou and Zhu [12] sought an instance segmentation of vehicles with pixel-level accuracy, where cars appear well delimited in a distinct physical instance. In this context, a severe problem is the differentiation of vehicles in contact that agglutinated in a single instance. The solution proposed by the authors was to establish an architecture that subdivided the central vehicle regions and their limits instead of treating the vehicle problem as a single unit. Reksten and Salberg [89] recently used the mask R-CNN with an image normalization strategy to suit different environments and an accurate road mask to filter driving vehicles from those parked.

Other studies combine a prior segmentation followed by vehicle detection. Audebert *et al.* [90] used the DL-based segment-before-detect method containing the following three steps:

1) semantic segmentation using a fully convolutional network to infer pixel-level class masks;
2) vehicle detection by regressing the bounding boxes of connected components;
3) object-level classification using CNN architectures (LeNet, AlexNet, and VGG-16).

TABLE II
RELATED WORKS USING OBJECT DETECTION ALGORITHMS, CONSIDERING THE METHOD AND DATA TYPE

| Paper | Method | Data |
|---|---|---|
| [62] | Hybrid Deep Convolutional Neural Network (HDNN) | 5 |
| [63] | Two step detection: BING to extract region proposals and feature extraction for classification with CNN | 1 |
| [64] | Two CNNs: AVPN to predict bounding boxes of the targets, and VALN for inferring type and orientation. | 2 |
| [65] | An improved vehicle detection method based on Faster R-CNN. | 3 |
| [50] | Vehicle detection using the Faster R-CNN | 3 |
| [66] | Method based on Cascaded Convolutional Neural Networks | 2 |
| [67] | Hard Example Mining (HEM) to the Stochastic Gradient Descent training of a CNN classifier. | 7 |
| [68] | Real-Time Ground Vehicle Detection based on CNN. | 3 |
| [69] | Development of the Deep Vehicle Counting Framework based on Enhanced-SSD | 4 |
| [70] | Comparison between YOLOv3 (best model) and Faster R-CNN | 3 |
| [71] | Detection model based on two CNNs that adopt the VGG-16 model | 2 |
| [72] | EOVNet (Earth observation image-based vehicle detection network), a modified Faster R-CNN. | 7 |
| [1] | Improved Faster R-CNN with Multiscale Feature Fusion and Homography Augmentation | 7 |
| [73] | R3-Net a deep network for multi-oriented vehicle detection | 2 |
| [74] | Detection algorithm based on Faster R-CNN | 2 |
| [75] | Systematic investigation of the Fast R-CNN and Faster R-CNN in vehicle detection | 2 |
| [5] | YOLOv3, vehicle tracking using deep appearance features, and Kalman filtering for motion estimation | 3 |
| [7] | Model based on multi-task cost-sensitive-convolutional neural network (MTCS-CNN) | 6 |
| [76] | Novel double focal loss convolutional neural network (DFLCNN) | 2 |
| [77] | Improved YOLOv3 using a sloping bounding box attached to the angle of the target vehicles | 2 |
| [78] | Orientation-aware feature fusion single-stage detection (OAFF-SSD) | 3 |
| [15] | Detection model for different scales using CNN and proposition of an Outlier-Aware Non-Maximum Suppression. | 3 |
| [79] | Comparison among faster R-CNN, R-FCN, and SSD (Best model) | 3 |
| [80] | Optimized DL model considering feature extraction, object detection, and non-maximum suppression. | 7 |
| [81] | Small-Sized Vehicle Detection Network (AVDNet) (one-stage vehicle detection network) | 2 |
| [82] | Comparison among four object detection networks: D-YOLO (best model), YOLOV2, YOLOV3, and YOLT | 1 |
| [83] | Vehicle detection based on RetinaNet architecture | 1 |
| [84] | Model based on Alexnet network (classification) and Faster R-CNN (target detection) | 1 |
| [85] | Faster R-CNN with a improved feature-balanced pyramid network (FBPN) | 2 |
| [86] | Comparison among YOLOv3, YOLOv4 (best models), and Faster R-CNN | 3 |
| [87] | Super-resolution cyclic GAN with RFA and YOLO as the detection network (SRCGAN-RFA-YOLO) | 1, 2 |
| [88] | Modified YOLOv3 and fcNN using 3D features in cascade. | 2 |
| [16] | Method using the lightweight feature extraction network with the Faster R-CNN | 2 |
| [17] | Orientation-Aware Vehicle Detection with an Anchor-Free Object Detection approach | 2 |

The data types are separated into seven categories: 1) satellite, 2) aerial, 3) UAV, (4) ultrahigh-resolution UAV, 5) google earth (GE), 6) cameras at the top of the building, and (7) several. Acronyms for the methods: residual feature aggregation (RFA), generative adversarial network (GAN), and YOLO.

Yu *et al.* [91] developed a convolutional capsule network with the following steps:

1) superpixel segmented;
2) labeling patches into vehicles or background using convolutional capsule network;
3) nonmaximum suppression to eliminate repetitive detections.

Tao *et al.* [92] performed a scene classification with DL followed by different vehicle detectors and postprocessing rules according to the scene context.

## III. MATERIAL AND METHODS

### A. Study Area and Image Acquisition

The entire city of Brasilia was the study area (see Fig. 2). Large regions with many mapped look-alike features and different scenarios favor learning DL models. The image has 57 856×42 496 spatial dimensions, and 0.24-m resolution obtained by the Infraestrutura de Dados Espaciais do Distrito Federal (IDE/DF).[1] In this scenario, a car has approximately 20 (length)×10 (width) pixel dimensions.

[1][Online]. Available: https://www.geoportal.seduh.df.gov.br/geoportal/, accessed on January 8, 2022.

### B. Semisupervised Iterative Learning

Manually identifying all the cars in a city is very time-consuming. So, the solution is to seek alternatives to automate the generation of datasets correctly. For example, if a very good annotator took five seconds to label a single car, it would take over 200 h to label 150 000 vehicles. Thus, we proposed a novel semisupervised approach using the geographic information system (GIS) data to increase operability (see Fig. 3). Briefly, the method consists in labeling a portion of the image for training the model and then using the model to classify the entire 57 856×42 496-pixel image. Then, we converted the predictions into the shapefile format easily edited in ArcMap, corrected the areas that present the most errors, and included them in the training data.

The proposed procedure to increase the training database reconciles incremental and cumulative learning, selecting samples that improve the model performance. An effective database expansion design aims to achieve greater incremental accuracy in subsequent predictions. The procedure is cumulative, using the entire set of labeled samples present in each step. Thus, the segmentation model increases its performance until the accuracy values do not vary significantly, i.e., the decrease in

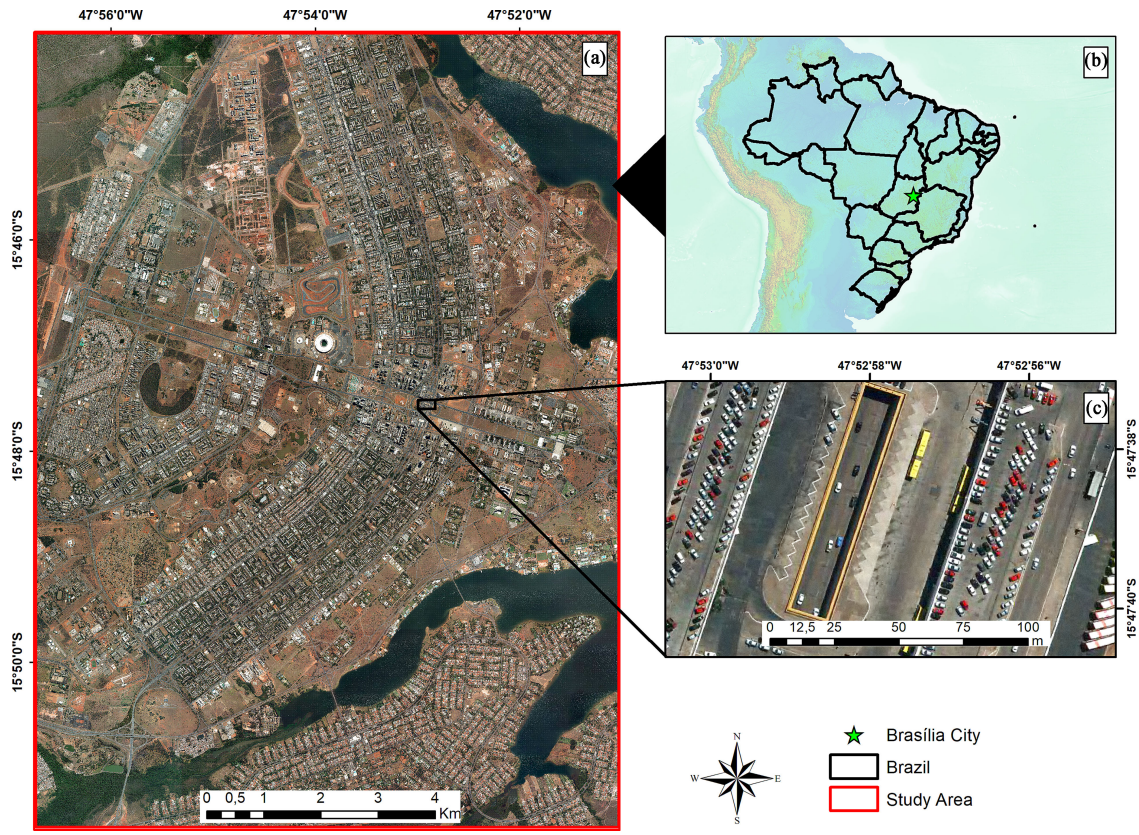


Fig. 2. Study area.

the incremental accuracy is due to the depletion of informative data.

*1) Ground Truth:* The manual annotations and corrections used the ArcMap software, considering a polygon shapefile for each vehicle since it is much easier to manipulate when compared to raster (mask) data. We applied a 1-pixel buffer (0.24 m in the corresponding image) with negative distance to generate the borders inside the polygon features. The first training procedure used training samples made from scratch. Subsequent iterations used the DL predictions as the primary raw data, with corrections for the areas with the most errors. The number of verified and corrected areas increases after each iteration using the semisupervised approach, increasing the dataset.

*2) DL Sample Generator Software:* The capture of DL samples must be in strategic areas. The present research proposed a novel method for selecting samples using the Point shapefile. This procedure allows choosing critical points where wrong predictions become part of new training after correction, quickly improving the model's detection capacity with much less laborious work. The developed DL sample generator from point shapefiles became a module in the Abilius Software program that receives the following three inputs:

1) the original image;
2) the ground truth image;
3) the point shapefiles.

The program requires inputs in the same projection, and the user may choose the size of the image tiles generated. The software uses the point shapefile to center the image tiles and crops the image and its corresponding ground truth image. Besides, this software outputs the annotations for instance segmentation, considering the COCO annotation format [14], which is compatible with region CNN methods [93], such as the mask-RCNN [58] and similar methods. Using point shapefiles also enables the user to generate samples close to each other, a powerful augmentation technique.

*3) DL Approach:* Usually, region-based instance segmentation underperforms on small objects, and semantic segmentation does not present distinct classification for different instances, unable to differentiate adjacent vehicles. The conversion of a conventional semantic segmentation model to a polygon shapefile with touching vehicles [see Fig. 4(a)] acquires a single polygon. Semantic segmentation models are the most used among the remote sensing community, mainly because of the good per-pixel results and simplicity of models and annotation formats. Thus, to solve this problem, we adopted a similar solution proposed by Mou and Zhu [12]. Instead of multitasking learning, we adopted a multiclass learning procedure in which the contour class competes against the vehicle class.

The model output subdivides the vehicle into two parts (edge and interior) [see Fig. 4(b)]. Deleting the edges isolates the individual vehicles, and all previously touching cars will be at least 2 pixels apart from each other. The next step is to develop a function to attribute a different value to each vehicle. This proposed method generates a list with all contours, using the OpenCV function (findContours) [94], and iteratively convert

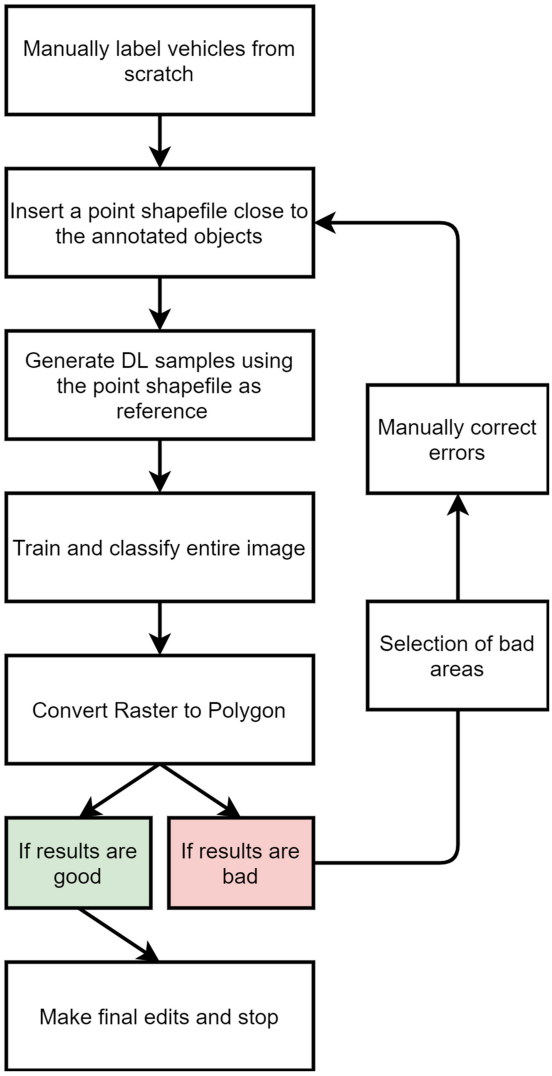


Fig. 3. Proposed semisupervised pipeline.

the contours to a mask attributing different values from 1 to N, being N the total number of distinct vehicles [see Fig. 4(c)]. Aiming to optimize computational resources, we adapted the polygon2mask function from the scikit-image package [95] that generates an array with zeros every time it is called, which is costly due to the enormous image dimensions. Thus, we only create an array with zeros once. In each iteration, we attribute different values to the generated mask (one object at a time), guaranteeing distinct values for each vehicle.

Now, the predictions are distinct for each object. However, since the objects are small, a 1-pixel error at the edges is considerable and not as precise. The edge restoration uses the instance array as the input. The first step is to apply 1-pixel padding in the entire image. Then, we make the following eight copies of the original array dislocated in different directions: 1) up, 2) down, 3) left, 4) right, 5) up-right, 6) up-left, 7) down-right, and 8) down-left. Then, we sum all arrays considering only pixels with zero value and remove the initial padding (recovering the image's original shape). This procedure enlarges the object edges, independent of the object orientation, resulting in the same semantic information [see Fig. 4(a)], but with different instances for each object [see Fig. 4(d)].

Despite the variety of semantic segmentation models, this study used a single combination throughout the iterative learning process since the primary goal is not to develop a new DL architecture but to make an efficient procedure for large areas per-pixel vehicle detection separating different instances. The configuration used the semantic segmentation models repository [96] and considered the U-net architecture [97] with the Efficient-net-B7 backbone [98]. Nevertheless, to present a more robust comparison, we evaluated the DeepLabv3+ [99], pyramid scene parsing network (PSPNet) [100], feature pyramid network (FPN) [101], and LinkNet [102] on final generated dataset, all of which using the Efficient-net-B7 backbone.

The hyperparameters were the same for all training iterations:

1) 300 epochs;
2) adam optimizer;
3) batch size of five.

Besides, the method considered the cross-entropy loss function with weights (0.1 for background, 0.6 for vehicles, 0.3 for the contour) and 15% of the images as validation, saving the model with the lowest cross-entropy loss. The dataset expansion used two augmentation strategies: the random horizontal and vertical flip, both with probabilities of 50%.

Moreover, we compared the proposed method with the mask-RCNN model [58] to evaluate the differences between a box-free method (ours) and a box-based method. In this context, the Detectron2 software is open source [103], being one of the most widely used in instance segmentation. It is important to state that there are limitations in comparing box-free and box-based methods because:

1) the hyperparameters are different;
2) the models are different (both architectures and backbones); and
3) the data format is different (e.g., instance segmentation models require data in the COCO annotation format).

The proposed annotation tool simultaneously provides semantic segmentation ground truth and COCO annotations for compatibility with box-based methods.

Three backbone configurations were tested (ResNeXt-101 [104], ResNet-101 [105], and ResNet-50), all of which presents pretrained weights, which speeds the training process. For box-based methods, a very substantial augmentation includes scaling the image dimensions, which increases the number of pixels for the object class, increasing results. In this regard, we considered two scenarios. The first considered the original image dimensions (256×256), and the second scenario scales the image to 1024×1024-pixel dimension. This augmentation strategy is much harder for semantic segmentation models (using our computer configurations, requiring a more robust GPU) since the computational cost would increase substantially, running out of memory. In contrast, the instance segmentation models allow this strategy since the segmentation masks are performed only for the proposed boxes. Despite the differences, the comparison is valid to understand if our proposed method is better at pixel-level accuracy, even using augmentations for the box-based instance segmentation methods that are not valid for

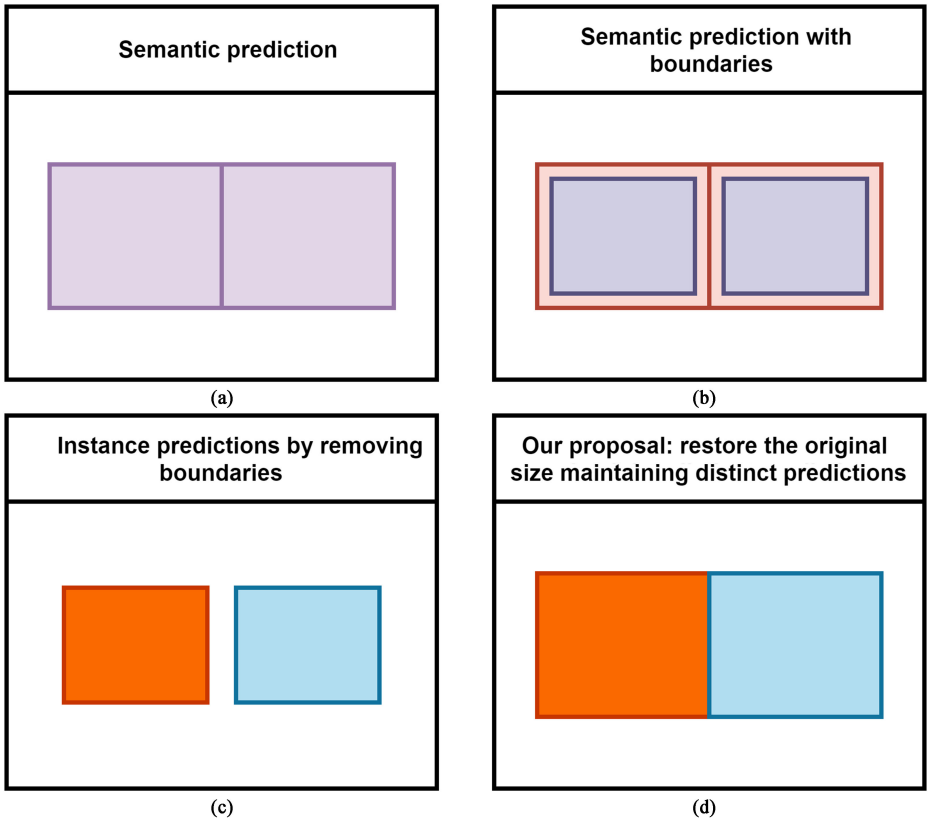


Fig. 4. Theoretical outputs from semantic segmentation algorithms, in which (a) is a normal semantic segmentation strategy, (b) is segmentation with boundaries, (c) is instance segmentation by removing the boundaries, and (d) is our proposed solution to restore the correct size maintaining distinct predictions. (a) Semantic prediction. (b) Semantic prediction with boundaries. (c) Instance predictions by removing boundaries. (d) Our proposal: restore the original size maintaining distinct predictions.

our proposed method. In both cases, we used random horizontal and vertical flips. The training used 10 000 iterations, two images per batch, and the other parameters as default.

*4) Large Image Classification:* The dimensions of the training images are 256×256, which is smaller than the entire image. Thus, we considered a sliding window approach with a 128-pixel stride to classify the whole image. The stride size smaller than the image dimensions results in overlapping pixels. A traditional way is to take the mean average among the overlapping pixels. Moreover, this approach reduces errors at the borders of the frames, exemplified in recent works [106]–[108]. A drawback of using this method is the computational cost. The time to classify an image increases nonlinear when reducing the stride value. Since our image presents large dimensions, we did not consider smaller stride values.

### C. Model Evaluation

The model evaluation considered a test set of 50 images with 256×256-pixel dimensions (same as dimensions for training and validation), and three independent testing areas (see Fig. 5), considering different difficulty scenarios. The first considered areas with no occlusion and significant difficulties for the cars [see Fig. 5(a)], with 2560×2560-pixel dimensions. The second scenario is a parking lot with many crowded vehicles [see Fig. 5(b)] with 2304×2304-pixel dimensions. The third scenario cover residential areas with a building generating shadow and regions of occlusion [see Fig. 5(c)] with 1560×1560-pixel dimensions. The semantic segmentation of the entire test area used a sliding window with 128-pixel steps. Meanwhile, the instance segmentation (mask-RCNN) of the testing areas used the mosaic method developed by Carvalho *et al.* [109].

In supervised learning tasks, the accuracy analysis compares the predicted results and the ground truth data. The confusion matrix is a standard structure for all tasks, yielding four possible outcomes: true positives (TP), true negatives, false positives (FP), and false negatives (FN). For semantic segmentation tasks, the confusion matrix analysis is per pixel. There are many possible metrics, such as overall accuracy, precision, recall, f-score, among others. Since we aim to evaluate how the metrics improve iteratively, we chose the intersection over union (IoU), which is widely adopted as one of the most important semantic segmentation metrics. The IoU is given by

$$\text{IoU} = \frac{|A \cap B|}{|A \cup B|} = \frac{\text{TP}}{\text{TP} + \text{FP} + \text{FN}}. \tag{1}$$

In which $A \cap B$ is the area of intersection, and $A \cup B$ is the area of union. The analysis considered the following: a) IoU for the test set and the three testing areas (considering the proposed expanded border algorithm and without considering the borders) at each iteration, and b) per-object metrics in the testing areas


Fig. 5. Zoom from the three separate testing areas A, B, and C.

(T1, T2, and T3). The object analysis had the following four classifications:

1) correct predictions;
2) partial predictions;
3) FP;
4) FN.

TABLE III
IoU Results for Our Proposed Method in the BSB Vehicle Dataset Considering the Expanded (exp.) Border Algorithm, and Not Considering the Borders, for Each Train Iteration

| Train # | Type | T1 | T2 | T3 | Test Set |
|---|---|---|---|---|---|
| 1 | No border | 63.19 | 63.67 | 51.97 | 52.60 |
| | Exp. Border | 80.80 | 77.23 | 66.89 | 66.03 |
| 2 | No border | 64.41 | 64.65 | 54.41 | 63.52 |
| | Exp. Border | 86.73 | 79.94 | 74.75 | 80.39 |
| 3 | No border | 60.40 | 62.27 | 52.43 | 61.49 |
| | Exp. Border | 87.69 | 82.31 | 75.95 | 80.73 |
| 4 | No border | 62.83 | 62.39 | 55.81 | 63.39 |
| | Exp. Border | 88.03 | 81.98 | 78.44 | 81.06 |
| 5 | No border | 63.98 | 64.13 | 56.24 | 64.51 |
| | Exp. Border | 88.37 | 81.31 | 77.10 | 82.45 |

## IV. Results

### A. Training Iterations

The final version of the dataset used a total of five iterations. The total number of point shapefiles was 1066 with training samples in various scenarios (see Fig. 6). Each iteration considered point shapefiles in areas where the errors did not disappear in previous iterations (to see if the mistakes disappeared). Still, at each iteration, the concentration of points had different focuses. For example, the second training focused on eliminating look-alike features, which already gives a good boost in performance metrics, with an easy correct the error, since we only need to delete some polygons. The fourth training had the minimum number of points since the areas required more corrections (e.g., parking lots), being more laborious. Thus, the proposed procedure effectively uses the results of the DL model in repeated corrections of pseudolabels. Gradually, the predictions become more reliable, minimizing errors and manual correction labor in each interaction.

### B. Metrics

*1) Pixel Metrics:* Table III lists the results for IoU on the four separate testing sets (Test Area 1, Test Area 2, Test Area 3, and Test Set), considering each training step. There is an evident rise in the metrics when increasing the number of training samples on the same independent test areas. Test Area 1 (T1) had the highest results, and it is indeed the easiest since there are no shadows and occluded cars. Test area 2 (T2) has a parking lot with many crowded vehicles, presenting more errors. Test Area 3 (T3) has many regions with shadows, and partial vehicles had the lowest IoU, bringing to light the difficulty in some areas, even for human specialists. The test set has fifty 256×256 samples all around the city, with varying difficulty levels. The IoU of the test set is approximately the average of the distinct testing areas (81.88).

Table IV lists the results considering different architectures using the Efficient-net-B7 backbone. For all models, the same behavior was still present, in which the expanded border algorithm had a higher value than without using the borders, showing that the method is not dependent on the model architecture used, but on the preparation of data. Besides, the PSPNet was by far the worst model, and the difference between the expanding border

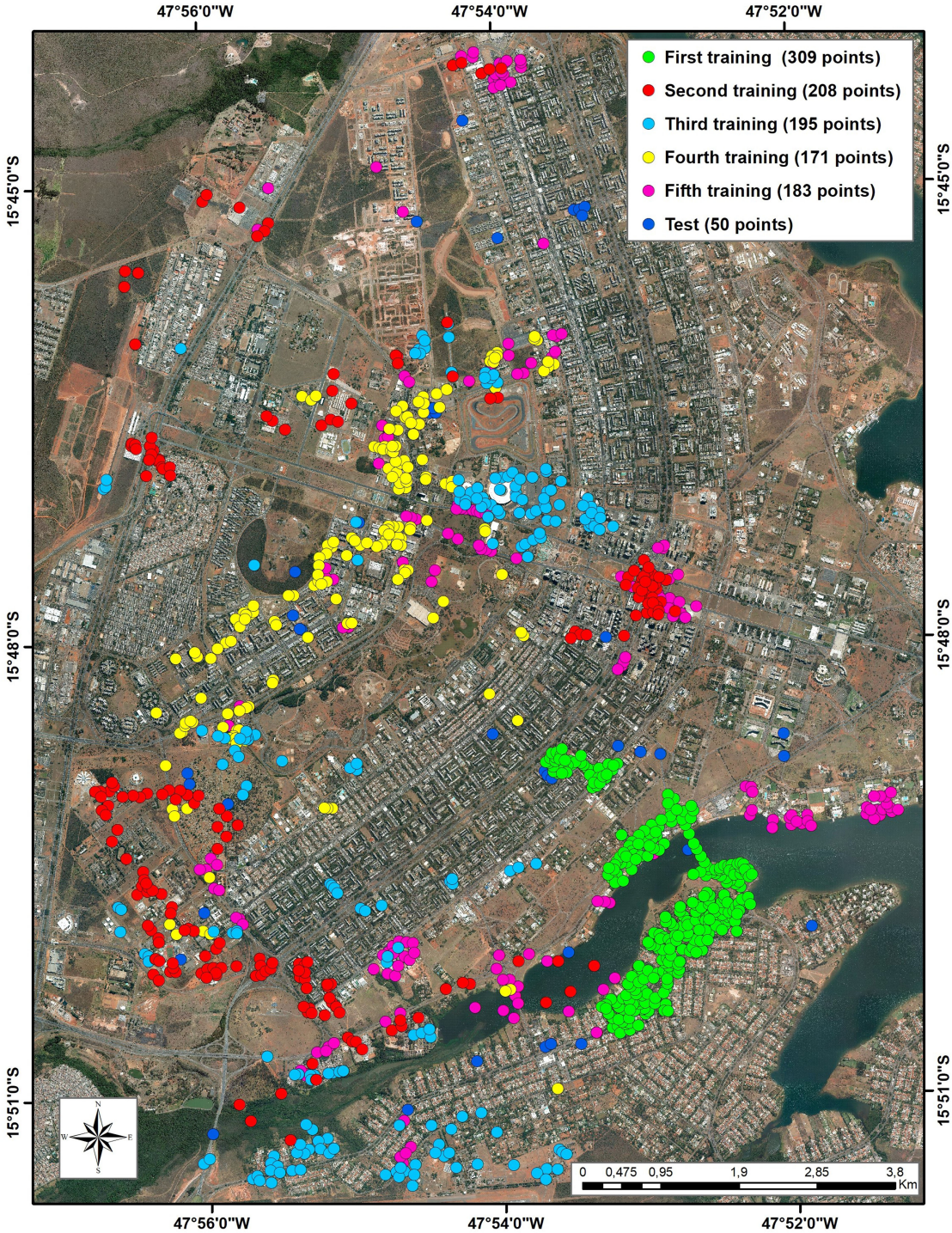


Fig. 6. Study area with the point shapefiles (training points) used in each training, in which the training is cumulative.

algorithm and without the borders was the lowest, showing that better models enhance the proposed algorithm even more. The DLv3+, LinkNet, and FPN presented slightly worse results than the U-net, demonstrating that the U-net was the best choice for this problem.

When comparing the IoU using our growing border algorithm to recover initial values without considering the borders, the results are very distinct, with a difference greater than 15% in the IoU metric. Also, the metrics remain very similar even when increasing the number of training samples. A possible explanation is error compensation, not bringing insightful information on the testing data.

Fig. 7 shows the semantic segmentation result, with and without borders. The visual results demonstrate that the proposed method expands vectorially 1 pixel on the edges, consisting of a fast process. Furthermore, the instances show an efficient separation. Fig. 7(b) (second row) demonstrates that the traditional predictions would merge the vehicles into a single polygon, if we

TABLE IV
IoU Results Considering the DeepLabv3+, LinkNet, PSPNet, and FPN Architectures Considering the Expanded (exp.) Border Algorithm, and Not Considering the Borders

| Model | Type | T1 | T2 | T3 | Test Set |
|---|---|---|---|---|---|
| DLv3+ | No border | 63.33 | 59.58 | 50.64 | 62.55 |
| | Exp. Border | 86.36 | 74.27 | 67.04 | 78.05 |
| LinkNet | No border | 66.47 | 63.50 | 53.73 | 64.93 |
| | Exp. Border | 86.78 | 79.33 | 70.31 | 81.31 |
| PSPNet | No border | 61.92 | 57.46 | 51.43 | 63.86 |
| | Exp. Border | 79.48 | 61.96 | 58.92 | 69.78 |
| FPN | No border | 62.83 | 62.39 | 55.81 | 63.10 |
| | Exp. Border | 88.03 | 81.98 | 78.44 | 78.26 |

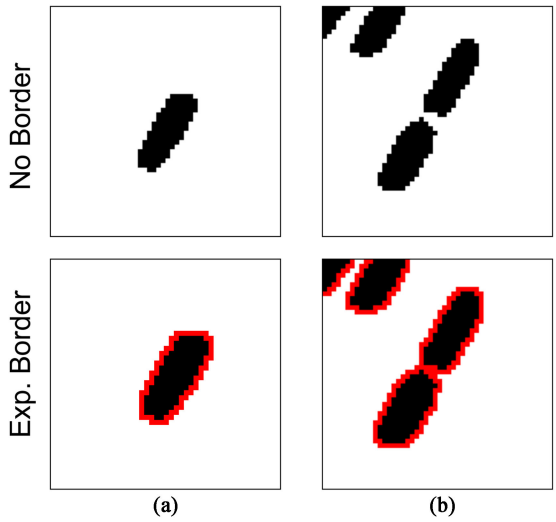


Fig. 7. Representation of two examples considering the vehicles with no borders and with expanded borders, in which the borders are highlighted in red.

TABLE V
IoU Results for the Mask-RCNN With ResNeXt-101 (X-101), ResNet-101 (R-101), and ResNet-50 (R-50) Backbones Considering Scaling Augmentation (1024×1024 Pixel Dimensions) and Without Scaling Augmentation (Original 256×256 Pixel Dimensions) in the BSB Vehicle Dataset

| backbone | scaling | T1 | T2 | T3 | Test Set |
|---|---|---|---|---|---|
| X-101 | Yes | 80.14 | 76.75 | 66.65 | 72.22 |
| | No | 75.41 | 63.88 | 55.17 | 67.06 |
| R-101 | Yes | 80.54 | 72.32 | 64.93 | 72.02 |
| | No | 76.13 | 65.01 | 55.51 | 65.80 |
| R-50 | Yes | 81.24 | 75.40 | 65.59 | 71.85 |
| | No | 79.40 | 66.59 | 55.32 | 66.49 |

have not differentiated them with the borders. Expanding edges on different instances retrieves the same semantic prediction information but with the distinction of the vehicles.

Table V lists the same testing areas but considers the mask-RCNN algorithm. Region algorithms rely on some procedures to enhance the classification of small objects. The results show that using the mask-RCNN with scaling the input image to 1024×1024 spatial dimensions (four times the original size) improves the results in more than 5% of IoU for all backbones. However, pixel metrics results are still far from the results using semantic segmentation architectures, in which the best model (ResNeXt-101) was more than 10% lower in IoU than the U-net model.

TABLE VI
Per Object Metrics: Correct Predictions (CP), Partial Predictions (PP), FN, and FP

| | T1 | T2 | T3 |
|---|---|---|---|
| CP | 89 | 395 | 430 |
| PP | 1 | 1 | 9 |
| FN | 0 | 5 | 21 |
| FP | 1 | 9 | 25 |

*2) Per Object Metrics:* Table VI lists the per object metrics (correct predictions, partial predictions, FN, and FP) on the three separate testing areas (T1, T2, and T3), considering the best model (containing all training samples). T1 classified all objects, showing that vehicles without shadows, occlusion, and crowded areas have very high precision. On the other hand, T3, with many shadow areas and occlusion, had the highest incidence of errors, with 21 FN and 25 FP. Considering that there were 430 correct predictions, the accuracy was still greater than 90%.

### C. Semantic to Instance Segmentation Results

Fig. 8 shows three zoomed areas considering the traditional semantic segmentation method (first two rows) and our proposed box-free instance segmentation method. Both figures consider the same model. The first row [see Fig. 8 (a), (b), and (c)] shows in yellow the merged cars, considering many vehicles in the same polygon, while the green cars were already independent even without our method. The second row (see Fig. 8(a1), (b1), and (c1)) shows the outlines of the polygons.

The third and fourth rows (Fig. 8(a2), (b2), (c2), (a3), (b3), and (c3)) show our proposed method considering the expanding border algorithm and separation into instance predictions. The fourth row shows cars in which each independent vector is represented by a different color, demonstrating that the method is efficient for separating vehicles in a precise pixel classification. Besides, interpreting these results gets much more straightforward, estimating the sizes of the vehicles and more accurate counting.

### D. Error Analysis

Even though the results were very accurate, some regions contain limitations. The training procedure used many look-alikes features to train a better model. However, the number of look-alikes in a city is extensive, introducing some mistakes (see Fig. 9 (b), (c), (e), and (f)). Some crowded areas may raise some errors by joining two cars (see Fig. 9 (a) and (d)).

### E. Final City-Scale Classification

The final city classification presented much fewer errors when compared to the first training. However, some errors were still present, as shown in the previous section. Fig. 10 shows the final classified image with a manual correction using two GIS specialists. The data are publicly available with 122 567 vehicles (car, bus, truck, and boat) [110].

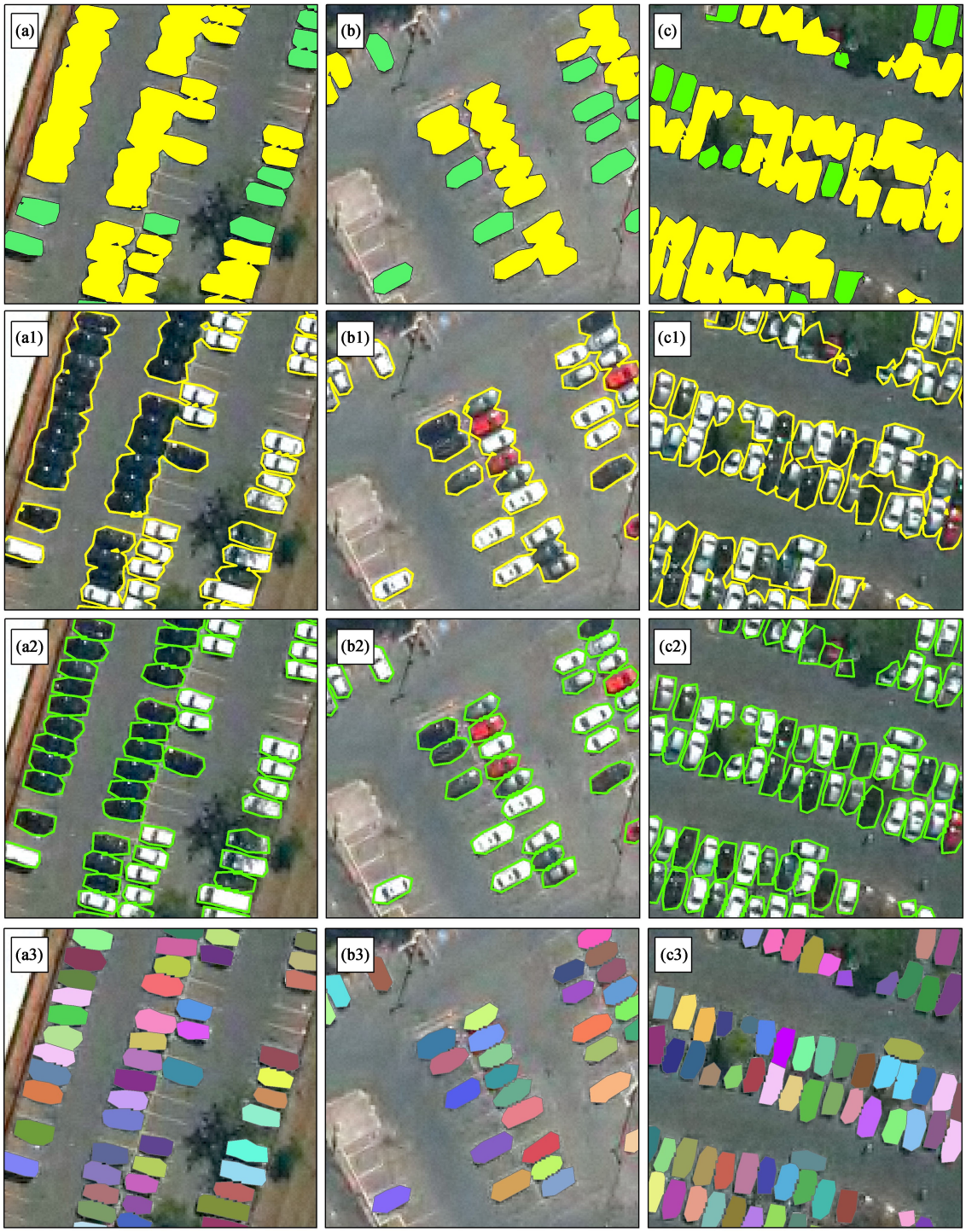


Fig. 8. Visual comparison of the traditional semantic segmentation results without using the border procedure (first two rows), and the proposed method (last two rows).

## V. Discussion

### A. Integration With GIS Software

To the best of authors' knowledge, this research is the first to use semisupervised iterative learning with GIS platform integration. We created a tool to generate the DL samples with corresponding ground truth data for semantic (PNG mask) and instance segmentation (COCO annotation format) to extract the best out of this method. A significant advantage of this method is understanding the misclassifications zones at each iteration, enabling choosing appropriate areas to continue the training with a substantial decrease in the laborious work. Besides, generating training samples from point shapefiles allows a dataset augmentation by selecting points in strategic regions, enabling the acquisition of many samples in a limited space. This iterative approach stays in hand with Koga *et al.*, supplying the algorithm


Fig. 9. Errors in the classification procedure, and errors present from the conversion from polygon to raster.

with complex examples (e.g., look-alikes). Our method allows obtaining the exact points in which the algorithm confuses with hard examples, being able to supply those mistaken areas back to training, rapidly improving results.

Moreover, the shapefile data is easy to manipulate, correct polygons, generate borders, change classes, among others, reducing problems, such as publicly available data with many errors in the ground truth data. Another great benefit is for end-users since the visualization of the data in those GIS platforms has many facilities, such as counting, choosing a specific area for analysis, getting the average size of the objects. Therefore, DL and GIS systems may work as allies for generating better predictions in less time.

### *B. Box-Free Instance Segmentation*

The instance segmentation results for vehicle mapping pursue the following two goals: 1) high separability between objects and 2) high per-pixel precision. The traditional instance segmentation models are region-based methods with a segmentation branch like the mask-RCNN. These box-based models have high object separability, but their pixel delimiting is lower than semantic segmentation models. Conversely, traditional semantic segmentation models cannot separate objects but have high per-pixel accuracy. Therefore, this study seeks a different approach from the traditional methods of instance segmentation, adapting the configuration of the input data and the image post-processing procedures to obtain, from semantic segmentation methods, results of the instances with greater precision. Thus, we proposed a box-free instance segmentation method using semantic segmentation models with object separation by turning the interiors of the borders into distinct polygons and restoring the original object size. The border approach accurately isolates the objects, making it easy to attribute unique values to each vehicle using nonlearning postprocessing steps. Mou and Zhu [12] had already introduced the usage of borders to separate instances. Even though the method is very interesting and effective, we incorporated the expanding border algorithm for more precise mapping. Our procedure uses a straightforward and fast vectorized approach to recover the 1-pixel at the borders of each object. In the literature, another proposal is by Tayara *et al.* [13], which uses dots to represent each car with a Gaussian elliptical shape, but the segmentation masks for each vehicle are ellipticals differing from the car shapes, applied only for counting.

The proposed box-free instance segmentation method demonstrated a competitive and superior performance than the mask R-CNN with different backbones and with and without image scaling. The application of image scaling is suitable for small objects (area $< 32^2$ pixels) [111], [112], such as cars, increasing their detection capability. In tests restricted to mask R-CNN, the best result considered ResNeXt-101 and image scaling to $1024 \times 1024$ pixel dimensions. However, the best mask R-CNN result was lower than our method using U-net with Efficientnet-B7 backbone (72% versus 82%). Therefore, the proposed method generated high-quality maps with distinct polygons for each object and presented a good pixel-wise accuracy, demonstrating adequation for this task. Besides, our proposed solution substitutes learning methods for object detection with nonlearning methods, which reduces the complexity of the entire process. For example, the mask-RCNN algorithm loss function is the sum of mask loss, classification loss, and box regression. In our proposed solution, we use a single loss function. Besides, we simplified the data preparation process, eliminating the

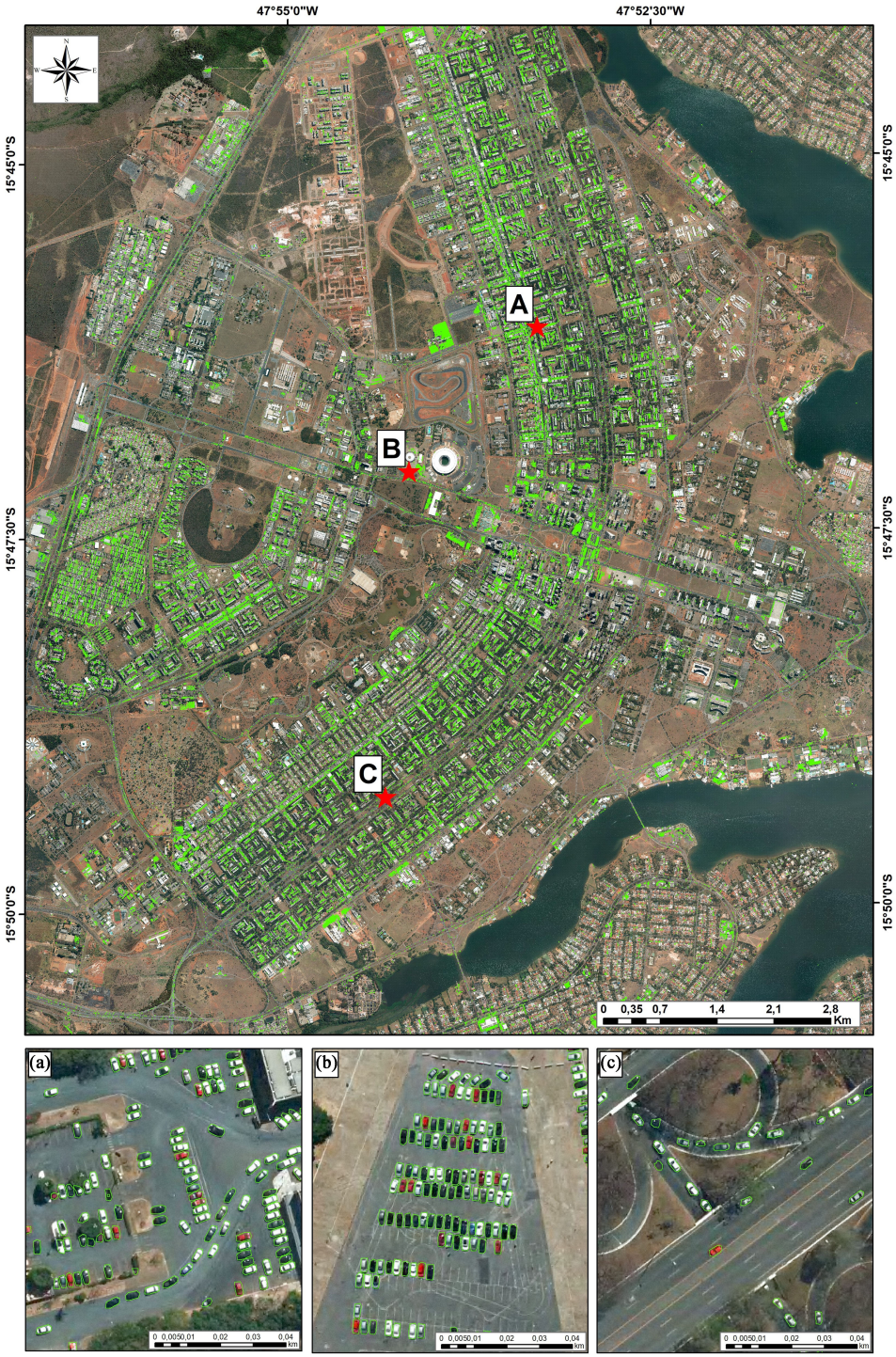


Fig. 10. Final image classification with three zoomed areas A, B, and C.

bounding boxes or storing any information in JSON files for use with other software. The training procedure only requires the image with its corresponding mask (with the borders). Thus, a simple change in the data preparation process allows the application of instance segmentation with more precise pixel-wise results.

The step of restoring the original object size by expanding its borders by 1 pixel is a crucial factor in increasing accuracy metrics, reaching 15% more IoU than without the edge regardless of the architecture tested. Considering that the cars in the analyzed images have a dimension of $20 \times 10$ pixels, a perfect prediction only limited to the interior would reach only 72% IoU. The better the model results, the greater the IoU differences between the result with and without the edge growth algorithm, see Table IV. These results imply the procedure of augmenting the vehicle dataset using iterative learning, which must consider the features with reconstituted edges to delineate the objects better and compensate for errors. In addition, the evaluation of

metrics per polygon in three test areas surpassed in all cases 90% in accuracy and recall. These results demonstrate an ideal scenario with good pixel mapping and the ability to distinguish different instances.

The large-area predictions using DL is an important topic that may be improved. Previous work shows that sliding windows with low step values correct errors at frame edges, improving results [90], [106]–[108]. It takes about one hour to classify our entire study area (57 856×42 496-pixel dimensions) using a 128-pixel stride. Future studies may evaluate the usage of parallel computing to accelerate this process.

This method can be easily adapted to other remote sensing targets (e.g., airplanes, buildings, houses, swimming pools). There is no need to use the borders for some targets that do not appear crowded, such as swimming pools, since the predictions will already be separated when extracting the polygons from the predicted mask. Besides, there are the following two possibilities for multiple targets at once: 1) create a new class for each contour, and 2) create a single contour class for all classes. In both cases, the loss function would remain the same. However, depending on how balanced the classes are, it might be necessary to use weights on each class. Besides, this methodology could be enhanced to fulfill other segmentation tasks, such as panoptic segmentation [113] in remote sensing datasets, such as the BSB aerial dataset [114].

### C. Vehicle Dataset

A promising trend in artificial intelligence considers data-centric approaches, which consists of leveraging the data quality. In the present research, we aimed for a precise pixel-wise classification maintaining different instances for each object, being very relevant for vehicle studies since most vehicle datasets aim to use object detection models (only bounding boxes) [115]–[118]. Some multiclass datasets also include vehicles [119], [120]. The iSAID dataset only comprises vehicles, for instance, segmentation tasks, with COCO annotation format annotations. Although object detection is very promising for counting vehicles, it requires adjustments (e.g., bounding box orientation) to obtain precise information (e.g., size), making the labeling procedure more complex. Moreover, to obtain pixel information about the cars to generate a map, it is crucial to get the boundaries of each object. Our proposed method can obtain pixel-wise instance-level predictions with the same information required for a traditional semantic segmentation model, a box-free method. Furthermore, our proposed dataset stores polygonal data, facilitating additional adjustments, such as dividing into more classes or refining labeled data.

Most vehicle studies use images with resolutions better than 20 cm. VAID [118] and VEDAI have the highest resolution (12.5 cm) among the data sets. Our dataset has a pixel resolution of 0.24 m, and the proposed method distinguished different instances, even at nearly twice the resolution of most datasets. The limitation of our dataset is that, for example, some distinguish sedans, which would be very difficult in our data. Therefore, our approach increases efficiency with a better resolution and is more suitable for separating into more classes (e.g., sedans, bus).

## VI. Conclusion

The present research presented the following three contributions:

1) a box-free instance segmentation method;
2) a semisupervised iterative approach to generate a high-quality dataset;
3) the BSB vehicle dataset.

The proposed DL method shows better results when compared to the mask-RCNN architecture with a pixel-wise IoU difference greater than 12%. We show that it is crucial to consider the borders for evaluating the pixel-wise mask, being very relevant to the proposed method to restore the objects' original size. The semisupervised iterative approach stabilized results in the fifth iteration, with a total of 1066 DL samples of 256×256 spatial dimensions. Our DL tool is a promising approach to generate datasets since it enables us to tackle strategic areas by inserting a point shapefile, significantly reducing laborious works. Finally, two specialists refined the BSB vehicle dataset containing more than 120 thousand unique vehicle polygons that are easily manipulative to other tasks.

The resolution in this research presents information very close to WorldView3 satellite imagery. Future research may consider the usage of more spectral bands in satellite data to enhance predictions. Besides, the results of our data are much better in situations without shadows and occlusion. For the generation of aerial imagery datasets, the researchers should consider training and evaluating the data in specific day periods with fewer shadows.

## References

[1] H. Ji, Z. Gao, T. Mei, and Y. Li, "Improved faster R-CNN with multiscale feature fusion and homography augmentation for vehicle detection in remote sensing images," *IEEE Geosci. Remote Sens. Lett.*, vol. 16, no. 11, pp. 1761–1765, Nov. 2019.

[2] V. K. Sakhare, T. Tewari, and V. Vyas, "Review of vehicle detection systems in advanced driver assistant systems," *Arch. Comput. Methods Eng.*, vol. 27, no. 2, pp. 591–610, 2020.

[3] D. Feng *et al.*, "Deep multi-modal object detection and semantic segmentation for autonomous driving: Datasets, methods, and challenges," *IEEE Trans. Intell. Transp. Syst.*, vol. 22, no. 3, pp. 1341–1360, Mar. 2021.

[4] J. Janai, F. Güney, A. Behl, and A. Geiger, "Computer vision for autonomous vehicles: Problems, datasets and state of the art," *Found. Trends Comput. Graph. Vis.*, vol. 12, no. 1–3, pp. 1–308, 2020.

[5] H. Wang, Y. Yu, Y. Cai, X. Chen, L. Chen, and Q. Liu, "A comparative study of state-of-the-art deep learning algorithms for vehicle detection," *IEEE Intell. Transp. Syst. Mag.*, vol. 11, no. 2, pp. 82–95, May/Jun. 2019.

[6] H. Song, H. Liang, H. Li, Z. Dai, and X. Yun, "Vision-based vehicle detection and counting system using deep learning in highway scenes," *Eur. Transport Res. Rev.*, vol. 11, no. 1, pp. 1–16, 2019.

[7] X. Xi, Z. Yu, Z. Zhan, Y. Yin, and C. Tian, "Multi-task cost-sensitive-convolutional neural network for car detection," *IEEE Access*, vol. 7, pp. 98061–98068, 2019.

[8] M. Fachrie *et al.*, "A simple vehicle counting system using deep learning with YOLOv3 model," *J. RESTI (Rekayasa Sistem dan Teknologi Informasi)*, vol. 4, no. 3, pp. 462–468, 2020.

[9] Z.-Q. Q. Zhao, P. Zheng, S.-T. T. Xu, and X. Wu, "Object detection with deep learning: A review," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 30, no. 11, pp. 3212–3232, Nov. 2019.

[10] Y. Guo, Y. Liu, T. Georgiou, and M. S. Lew, "A review of semantic segmentation using deep neural networks," *Int. J. Multimedia Inf. Retrieval*, vol. 7, no. 2, pp. 87–93, 2018.

[11] A. M. Hafiz and G. M. Bhat, "A survey on instance segmentation: State of the art," *Int. J. Multimedia Inf. Retrieval*, vol. 9, no. 3, pp. 171–189, 2020.

[12] L. Mou and X. X. Zhu, "Vehicle instance segmentation from aerial image and video using a multitask learning residual fully convolutional network," *IEEE Trans. Geosci. Remote Sens.*, vol. 56, no. 11, pp. 6699–6711, Nov. 2018.

[13] H. Tayara, K. Gil Soo, and K. T. Chong, "Vehicle detection and counting in high-resolution aerial images using convolutional regression neural network," *IEEE Access*, vol. 6, pp. 2220–2230, 2018.

[14] T.-Y. Lin *et al.*, "Microsoft COCO: Common objects in context," in *Proc. Eur. Conf. Comput. Vis.*, 2014, pp. 740–755.

[15] X. Li, X. Li, and H. Pan, "Multi-scale vehicle detection in high-resolution aerial images with context information," *IEEE Access*, vol. 8, pp. 208643–208657, 2020.

[16] J. Shen, N. Liu, and H. Sun, "Vehicle detection in aerial images based on lightweight deep convolutional network," *IET Image Process.*, vol. 15, no. 2, pp. 479–491, 2021.

[17] F. Shi, T. Zhang, and T. Zhang, "Orientation-aware vehicle detection in aerial images via an anchor-free object detection approach," *IEEE Trans. Geosci. Remote Sens.*, vol. 59, no. 6, pp. 5221–5233, Jun. 2021.

[18] S. Hinz, "Detection and counting of cars in aerial images," in *Proc. Int. Conf. Image Process.*, vol. 3, 2003, pp. III-997–1000.

[19] G. Sharma, C. J. Merry, P. Goel, and M. McCord, "Vehicle detection in 1-m resolution satellite and airborne imagery," *Int. J. Remote Sens.*, vol. 27, no. 4, pp. 779–797, 2006.

[20] Z. Zheng *et al.*, "A novel vehicle detection method with high resolution highway aerial image," *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 6, no. 6, pp. 2338–2343, Dec. 2013.

[21] H. Y. Cheng, C. C. Weng, and Y. Y. Chen, "Vehicle detection in aerial surveillance using dynamic Bayesian networks," *IEEE Trans. Image Process.*, vol. 21, no. 4, pp. 2152–2159, Apr. 2012.

[22] M. D. Hossain and D. Chen, "Segmentation for object-based image analysis (OBIA): A review of algorithms and challenges from remote sensing perspective," *ISPRS J. Photogramm. Remote Sens.*, vol. 150, pp. 115–134, 2019.

[23] A. C. Holt, E. Y. Seto, T. Rivard, and P. Gong, "Object-based detection and classification of vehicles from high-resolution aerial photography," *Photogrammetric Eng. Remote Sens.*, vol. 75, no. 7, pp. 871–880, 2009.

[24] L. Eikvil, L. Aurdal, and H. Koren, "Classification-based vehicle detection in high-resolution satellite images," *ISPRS J. Photogramm. Remote Sens.*, vol. 64, no. 1, pp. 65–72, 2009.

[25] Z. Chen *et al.*, "Vehicle detection in high-resolution aerial images via sparse representation and superpixels," *IEEE Trans. Geosci. Remote Sens.*, vol. 54, no. 1, pp. 103–116, Jan. 2016.

[26] N. Dalal and B. Triggs, "Histograms of oriented gradients for human detection," in *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.*, 2005, vol. 1, no. 7, pp. 886–893.

[27] P. Viola and M. Jones, "Rapid object detection using a boosted cascade of simple features," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2001, vol. 1, pp. I-511–I-518.

[28] T. Moranduzzo and F. Melgani, "Automatic car counting method for unmanned aerial vehicle images," *IEEE Trans. Geosci. Remote Sens.*, vol. 52, no. 3, pp. 1635–1647, Mar. 2014.

[29] T. Moranduzzo and F. Melgani, "Detecting cars in UAV images with a catalog-based approach," *IEEE Trans. Geosci. Remote Sens.*, vol. 52, no. 10, pp. 6356–6367, Oct. 2014.

[30] J. Leitloff, D. Rosenbaum, F. Kurz, O. Meynberg, and P. Reinartz, "An operational system for estimating road traffic information from aerial images," *Remote Sens.*, vol. 6, no. 11, pp. 11315–11341, 2014.

[31] T. T. Nguyen, H. Grabner, H. Bischof, and B. Gruber, "On-line boosting for car detection from aerial images," in *Proc. IEEE Int. Conf. Res., Innov. Vis. Future*, 2007, pp. 87–95.

[32] S. Tuermer, F. Kurz, P. Reinartz, and U. Stilla, "Airborne vehicle detection in dense urban areas using hog features and disparity maps," *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 6, no. 6, pp. 2327–2337, Dec. 2013.

[33] X. Cao, C. Wu, J. Lan, P. Yan, and X. Li, "Vehicle detection and motion analysis in low-altitude airborne video under urban environment," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 21, no. 10, pp. 1522–1533, Oct. 2011.

[34] J. Han, D. Zhang, G. Cheng, L. Guo, and J. Ren, "Object detection in optical remote sensing images based on weakly supervised learning and high-level feature learning," *IEEE Trans. Geosci. Remote Sens.*, vol. 53, no. 6, pp. 3325–3337, Jun. 2015.

[35] Y. Yu, H. Guan, and Z. Ji, "Rotation-invariant object detection in high-resolution satellite imagery using superpixel-based deep hough forests," *IEEE Geosci. Remote Sens. Lett.*, vol. 12, no. 11, pp. 2183–2187, Nov. 2015.

[36] Y. Yu, H. Guan, D. Zai, and Z. Ji, "Rotation-and-scale-invariant airplane detection in high-resolution satellite images based on deep-hough-forests," *ISPRS J. Photogramm. Remote Sens.*, vol. 112, pp. 50–64, 2016.

[37] L. Cao, Q. Jiang, M. Cheng, and C. Wang, "Robust vehicle detection by combining deep features with exemplar classification," *Neurocomputing*, vol. 215, pp. 225–231, 2016.

[38] N. Ammour, H. Alhichri, Y. Bazi, B. Benjdira, N. Alajlan, and M. Zuair, "Deep learning approach for car detection in UAV imagery," *Remote Sens.*, vol. 9, no. 4, p. 312, 2017.

[39] F. Leberl, H. Bischof, H. Grabner, and S. Kluckner, "Recognizing cars in aerial imagery to improve orthophotos," in *Proc. ACM Int. Symp. Adv. Geographic Inf. Syst.*, 2007, pp. 2–10.

[40] H. Grabner, T. T. Nguyen, B. Gruber, and H. Bischof, "On-line boosting-based car detection from aerial images," *ISPRS J. Photogramm. Remote Sens.*, vol. 63, no. 3, pp. 382–396, 2008.

[41] J. Gleason, V. A. Nefian, X. Bouyssounousse, T. Fong, and G. Bebis, "Vehicle detection from aerial imagery," in *Proc. IEEE Int. Conf. Robot. Autom.*, 2011, pp. 2065–2070.

[42] A. Kembhavi, D. Harwood, and L. S. Davis, "Vehicle detection using partial least squares," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 33, no. 6, pp. 1250–1265, Jun. 2011.

[43] P. Liang, G. Teodoro, H. Ling, E. Blasch, G. Chen, and L. Bai, "Multiple kernel learning for vehicle detection in wide area motion imagery," in *Proc. IEEE 15th Int. Conf. Inf. Fusion*, 2012, pp. 1629–1636.

[44] W. Shao, W. Yang, G. Liu, and J. Liu, "Car detection from high-resolution aerial imagery using multiple features," in *Proc. IEEE Int. Geosci. Remote Sens. Symp.*, 2012, pp. 4379–4382.

[45] K. Liu and G. Mattyus, "Fast multiclass vehicle detection on aerial images," *IEEE Geosci. Remote Sens. Lett.*, vol. 12, no. 9, pp. 1938–1942, Sep. 2015.

[46] S. Madhogaria, P. Baggenstoss, M. Schikora, W. Koch, and D. Cremers, "Car detection by fusion of hog and causal MRF," *IEEE Trans. Aerosp. Electron. Syst.*, vol. 51, no. 1, pp. 575–590, Jan. 2015.

[47] S. Razakarivony and F. Jurie, "Vehicle detection in aerial imagery: A small target detection benchmark," *J. Vis. Commun. Image Representation*, vol. 34, pp. 187–203, 2016.

[48] Y. Xu, G. Yu, Y. Wang, X. Wu, and Y. Ma, "A hybrid vehicle detection method based on Viola-Jones and hog SVM from UAV images," *Sensors*, vol. 16, no. 8, p. 1325, 2016.

[49] L. Cao *et al.*, "Weakly supervised vehicle detection in satellite images via multi-instance discriminative learning," *Pattern Recognit.*, vol. 64, pp. 417–424, 2017.

[50] Y. Xu, G. Yu, Y. Wang, X. Wu, and Y. Ma, "Car detection from low-altitude UAV imagery with the faster R-CNN," *J. Adv. Transp.*, vol. 2017, 2017, Art. no. 2823617.

[51] H. Zhou, L. Wei, C. P. Lim, D. Creighton, and S. Nahavandi, "Robust vehicle detection in aerial images using bag-of-words and orientation aware scanning," *IEEE Trans. Geosci. Remote Sens.*, vol. 56, no. 12, pp. 7074–7085, Dec. 2018.

[52] C. Liu, Y. Ding, M. Zhu, J. Xiu, M. Li, and Q. Li, "Vehicle detection in aerial images using a fast oriented region search and the vector of locally aggregated descriptors," *Sensors*, vol. 19, no. 15, p. 3294, 2019.

[53] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," *Commun. ACM*, vol. 60, no. 6, pp. 84–90, 2017.

[54] I. Sevo and A. Avramovic, "Convolutional neural network based automatic object detection on aerial images," *IEEE Geosci. Remote Sens. Lett.*, vol. 13, no. 5, pp. 740–744, May 2016.

[55] R. Girshick, J. Donahue, T. Darrell, and J. Malik, "Rich feature hierarchies for accurate object detection and semantic segmentation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2014, vol. 1, pp. 580–587.

[56] R. Girshick, "Fast R-CNN," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2015, pp. 1440–1448.

[57] S. Ren, K. He, R. Girshick, and J. Sun, "Faster R-CNN: Towards real-time object detection with region proposal networks," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 39, no. 6, pp. 1137–1149, Jun. 2017.

[58] K. He, G. Gkioxari, P. Dollar, and R. Girshick, "Mask R-CNN," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 42, no. 2, pp. 386–397, Feb. 2020.

[59] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You only look once: Unified, real-time object detection," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2016, pp. 779–788.

[60] A. Van Etten, "You only look twice: Rapid multi-scale object detection in satellite imagery," 2018, *arXiv:1805.09512*.

[61] W. Liu *et al.*, "SSD: Single shot multibox detector," in *Proc. Eur. Conf. Comput. Vis.*, 2016, pp. 21–37.

[62] X. Chen, S. Xiang, C. L. Liu, and C. H. Pan, "Vehicle detection in satellite images by hybrid deep convolutional neural networks," *IEEE Geosci. Remote Sens. Lett.*, vol. 11, no. 10, pp. 1797–1801, Oct. 2014.

[63] S. Qu, Y. Wang, G. Meng, and C. Pan, "Vehicle detection in satellite images by incorporating objectness and convolutional neural network," *J. Ind. Intell. Inf.*, vol. 4, no. 2, pp. 158–162, 2016.

[64] Z. Deng, H. Sun, S. Zhou, J. Zhao, and H. Zou, "Toward fast and accurate vehicle detection in aerial images using coupled region-based convolutional neural networks," *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 10, no. 8, pp. 3652–3664, Aug. 2017.

[65] T. Tang, S. Zhou, Z. Deng, H. Zou, and L. Lei, "Vehicle detection in aerial images based on region convolutional neural networks and hard negative example mining," *Sensors*, vol. 17, no. 2, p. 336, 2017.

[66] J. Zhong, T. Lei, and G. Yao, "Robust vehicle detection in aerial images based on cascaded convolutional neural networks," *Sensors*, vol. 17, no. 12, 2017, Art. no. 2720.

[67] Y. Koga, H. Miyazaki, and R. Shibasaki, "A CNN-based method of vehicle detection from aerial images using hard example mining," *Remote Sens.*, vol. 10, no. 1, 2018, Art. no. 124.

[68] X. Liu, T. Yang, and J. Li, "Real-time ground vehicle detection in aerial infrared imagery based on convolutional neural network," *Electron. (Switzerland)*, vol. 7, no. 6, pp. 1–19, 2018.

[69] J. Zhu *et al.*, "Urban traffic density estimation based on ultrahigh-resolution UAV video and deep neural network," *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 11, no. 12, pp. 4968–4981, Dec. 2018.

[70] B. Benjdira, T. Khursheed, A. Koubaa, A. Ammar, and K. Ouni, "Car detection using unmanned aerial vehicles: Comparison between faster R-CNN and YOLOv3," in *Proc. 1st Int. Conf. Unmanned Veh. Syst.-Oman*, 2019, pp. 1–6.

[71] C. Chen, J. Zhong, and Y. Tan, "Multiple-oriented and small object detection with convolutional neural networks for aerial image," *Remote Sens.*, vol. 11, no. 18, p. 2176, 2019.

[72] Z. Gao, H. Ji, T. Mei, B. Ramesh, and X. Liu, "Eovnet: Earth-observation image-based vehicle detection network," *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 12, no. 9, pp. 3552–3561, Sep. 2019.

[73] Q. Li, L. Mou, Q. Xu, Y. Zhang, and X. X. Zhu, "$R^3$-net: A deep network for multioriented vehicle detection in aerial images and videos," *IEEE Trans. Geosci. Remote Sens.*, vol. 57, no. 7, pp. 5028–5042, Jul. 2019.

[74] J. Shen, N. Liu, and H. Sun, "Vehicle detection in aerial images based on hyper feature map in deep convolutional network," *KSII Trans. Internet Inf. Syst.*, vol. 13, no. 4, pp. 479–491, 2019.

[75] L. Sommer, T. Schuchert, and J. Beyerer, "Comprehensive analysis of deep learning-based vehicle detection in aerial images," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 29, no. 9, pp. 2733–2747, Sep. 2019.

[76] M. Y. Yang, W. Liao, X. Li, Y. Cao, and B. Rosenhahn, "Vehicle detection in aerial images," *Photogrammetric Eng. Remote Sens.*, vol. 85, no. 4, pp. 297–304, 2019.

[77] X. Zhang and X. Zhu, "An efficient and scene-adaptive algorithm for vehicle detection in aerial images using an improved YOLOv3 framework," *ISPRS Int. J. Geo- Inf.*, vol. 8, no. 11, p. 483, 2019.

[78] Y. Guo, Y. Xu, and S. Li, "Dense construction vehicle detection based on orientation-aware feature fusion convolutional neural network," *Autom. Construction*, vol. 112, 2019, Art. no. 103124.

[79] S. W. Ham, H. C. Park, E. J. Kim, S. Y. Kho, and D. K. Kim, "Investigating the influential factors for practical application of multi-class vehicle detection for images from unmanned aerial vehicle using deep learning models," *Transp. Res. Rec.*, vol. 2674, no. 12, pp. 553–567, 2020.

[80] S. Jiang *et al.*, "An optimized deep neural network detecting small and narrow rectangular objects in Google earth images," *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 13, pp. 1068–1081, 2020.

[81] M. Mandal, M. Shah, P. Meena, S. Devi, and S. K. Vipparthi, "AVD-Net: A small-sized vehicle detection network for aerial visual data," *IEEE Geosci. Remote Sens. Lett.*, vol. 17, no. 3, pp. 494–498, Mar. 2020.

[82] T. Ophoff, S. Puttemans, V. Kalogirou, J.-P. Robin, and T. Goedemé, "Vehicle and vessel detection on satellite imagery: A comparative study on single-shot detectors," *Remote Sens.*, vol. 12, no. 7, p. 1217, 2020.

[83] D. G. Stuparu, R. I. Ciobanu, and C. Dobre, "Vehicle detection in overhead satellite images using a one-stage object detection model," *Sensors (Switzerland)*, vol. 20, no. 22, pp. 1–18, 2020.

[84] Q. Tan, J. Ling, J. Hu, X. Qin, and J. Hu, "Vehicle detection in high resolution satellite remote sensing images based on deep learning," *IEEE Access*, vol. 8, pp. 153394–153402, 2020.

[85] B. Wang and Y. Gu, "An improved FBPN-based detection network for vehicles in aerial images," *Sensors (Switzerland)*, vol. 20, no. 17, pp. 1–20, 2020.

[86] A. Ammar, A. Koubaa, M. Ahmed, A. Saad, and B. Benjdira, "Vehicle detection from aerial images using deep learning: A comparative study," *Electron. (Switzerland)*, vol. 10, no. 7, pp. 1–31, 2021.

[87] S. M. A. Bashir and Y. Wang, "Small object detection in remote sensing images with residual feature aggregation-based super-resolution and object detector network," *Remote Sens.*, vol. 13, no. 9, p. 1854, 2021.

[88] S. Javadi, M. Dahl, and M. I. Pettersson, "Vehicle detection in aerial images based on 3D depth maps and deep neural networks," *IEEE Access*, vol. 9, pp. 8381–8391, 2021.

[89] J. H. Reksten and A. B. Salberg, "Estimating traffic in urban areas from very-high resolution aerial images," *Int. J. Remote Sens.*, vol. 42, no. 3, pp. 865–883, 2021.

[90] N. Audebert *et al.*, "Deep learning for urban remote sensing," in *Proc. Joint Urban Remote Sens. Event*, 2017, pp. 1–4.

[91] Y. Yu, T. Gu, H. Guan, D. Li, and S. Jin, "Vehicle detection from high-resolution remote sensing imagery using convolutional capsule networks," *IEEE Geosci. Remote Sens. Lett.*, vol. 16, no. 12, pp. 1894–1898, Dec. 2019.

[92] C. Tao, L. Mi, Y. Li, J. Qi, Y. Xiao, and J. Zhang, "Scene context-driven vehicle detection in high-resolution aerial images," *IEEE Trans. Geosci. Remote Sens.*, vol. 57, no. 10, pp. 7339–7351, Oct. 2019.

[93] R. Girshick, J. Donahue, T. Darrell, and J. Malik, "Region-based convolutional networks for accurate object detection and segmentation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 38, no. 1, pp. 142–158, Jan. 2016.

[94] G. Bradski and A. Kaehler, *Learning OpenCV: Computer Vision With the OpenCV Library*. Newton, MA, USA: O'Reilly Media, Inc., 2008.

[95] S. van der Walt *et al.*, "Scikit-image: Image processing in python," *PeerJ*, vol. 2, 2014, Art. no. e453.

[96] P. Yakubovskiy, *Segmentation Models Pytorch*, GitHub Repository, San Francisco, CA, USA, 2020.

[97] O. Ronneberger, P. Fischer, and T. Brox, "U-Net: Convolutional networks for biomedical image segmentation," in *Int. Conf. Med. Image Comput. Comput.-Assist. Interv.*, Cham, Switzerland: Springer, 2015, pp. 234–241.

[98] M. Tan and V. Q. Le, "Efficientnet: Rethinking model scaling for convolutional neural networks," in *Proc. Int. Conf. Mach. Learn.*, 2019, pp. 6105–6114.

[99] L.-C. Chen, G. Papandreou, F. Schroff, and H. Adam, "Rethinking atrous convolution for semantic image segmentation," 2017, *arXiv:1706.05587*.

[100] H. Zhao, J. Shi, X. Qi, X. Wang, and J. Jia, "Pyramid scene parsing network," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2017, pp. 2881–2890.

[101] T.-Y. Lin, P. Dollár, R. Girshick, K. He, B. Hariharan, and S. Belongie, "Feature pyramid networks for object detection," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2017, pp. 2117–2125.

[102] A. Chaurasia and E. Culurciello, "Linknet: Exploiting encoder representations for efficient semantic segmentation," in *Proc. IEEE Vis. Commun. Image Process.*, 2017, pp. 1–4.

[103] Y. Wu, A. Kirillov, F. Massa, W.-Y. Lo, and R. Girshick, "Detectron2," 2019. Accessed: Mar. 3, 2021. [Online]. Available: https://github.com/facebookresearch/detectron2

[104] S. Xie, R. Girshick, P. Dollar, Z. Tu, and K. He, "Aggregated residual transformations for deep neural networks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2017, pp. 5987–5995.

[105] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2016, pp. 770–778.

[106] A. O. de Albuquerque *et al.*, "Deep semantic segmentation of center pivot irrigation systems from remotely sensed data," *Remote Sens.*, vol. 12, no. 13, p. 2159, 2020.

[107] L. B. da Costa, O. L. F. de Carvalho, A. O. de Albuquerque, R. A. T. Gomes, R. F. Guimarães, and O. A. de Carvalho Júnior, "Deep semantic segmentation for detecting eucalyptus planted forests in the Brazilian territory using sentinel-2 imagery," *Geocarto Int.*, pp. 1–13, 2021, [Online]. Available. https://www.tandfonline.com/doi/abs/10.1080/10106049.2021.1943009

[108] M. V. C. V. D. Costa *et al.*, "Remote sensing for monitoring photovoltaic solar plants in Brazil using deep semantic segmentation," *Energies*, vol. 14, no. 10, p. 2960, 2021.
[109] O. L. F. de. Carvalho *et al.*, "Instance segmentation for large, multi-channel remote sensing imagery using mask-RCNN and a mosaicking approach," *Remote Sens.*, vol. 13, no. 1, p. 39, 2021.
[110] O. Carvalho *et al.*, "BSB vehicle dataset," 2022. [Online]. Available: https://figshare.com/articles/dataset/BSB_Vehicle_Dataset/18092822/1
[111] O. L. F. de Carvalho *et al.*, "Instance segmentation for governmental inspection of small touristic infrastructure in beach zones using multi-spectral high-resolution worldview-3 imagery," *ISPRS Int. J. Geo- Inf.*, vol. 10, no. 12, p. 813, 2021.
[112] K. Tong, Y. Wu, and F. Zhou, "Recent advances in small object detection based on deep learning: A review," *Image Vis. Comput.*, vol. 97, 2020, Art. no. 103910.
[113] A. Kirillov, K. He, R. Girshick, C. Rother, and P. Dollár, "Panoptic segmentation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2019, pp. 9404–9413.
[114] O. L. F. de Carvalho *et al.*, "Panoptic segmentation meets remote sensing," *Remote Sens.*, vol. 14, no. 4, p. 965, 2022.
[115] S. Drouyer, "Vehsat: A large-scale dataset for vehicle detection in satellite images," in *Proc. IEEE Int. Geosci. Remote Sens. Symp.*, 2020, pp. 268–271.
[116] Y. Zeng, Q. Duan, X. Chen, D. Peng, Y. Mao, and K. Yang, "Uavdata: A dataset for unmanned aerial vehicle detection," *Soft Comput.*, vol. 25, no. 7, pp. 5385–5393, 2021.
[117] S. M. Azimi, R. Bahmanyar, C. Henry, and F. Kurz, "Eagle: Large-scale vehicle detection dataset in real-world scenarios using aerial imagery," in *Proc. Int. Conf. Pattern Recognit.*, 2021, pp. 6920–6927.
[118] H.-Y. Lin, K.-C. Tu, and C.-Y. Li, "Vaid: An aerial image dataset for vehicle detection and classification," *IEEE Access*, vol. 8, pp. 212209–212219, 2020.
[119] S. W. Zamir *et al.*, "Isaid: A large-scale dataset for instance segmentation in aerial images," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. Workshops*, Long Beach, CA, USA, 2019, pp. 28–37.
[120] G.-S. Xia *et al.*, "DOTA: A large-scale dataset for object detection in aerial images," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2018, pp. 3974–3983.

**Osmar Luiz Ferreira de Carvalho** (Member, IEEE) received the B.Sc. degree in electrical engineering in 2020 from the University of Brasília, Brasília, Brazil, where he is currently working toward the M.Sc. degree in computer science.

He is currently a Machine Learning Engineer with the industry and research projects.

Dr. Carvalho was a founding member of the IEEE Computer Intelligence Society Student Chapter at the University of Brasília, winning the Outstanding Chapter of the Year Award in 2019, during his undergraduate.

**Osmar Abílio de Carvalho Júnior** (Member, IEEE) received the B.Sc., M.Sc., and Ph.D. degrees in geology from the University of Brasília, Brasília, Brazil, 1990, 1995, and 2000, respectively.

He is currently an Associate Professor with the University of Brasília with broad experience in scientific projects. His main research interest includes remote sensing, GIS, software development, computer vision, and image classification.

**Anesmar Olino de Albuquerque** (Student Member, IEEE) received the B.Sc. degree in geography from the State University of Goiás, Goiás, Brazil, and the M.Sc. and Ph.D. degrees in geography from the University of Brasília, Brasília, Brazil, in 2009, 2015, and 2022, respectively.

He is currently a Researcher with the Laboratory of Space Information and Systems, University of Brasília.

**Nickolas Castro Santana** (Member, IEEE) received the B.Sc. degree in geography from the University Center of Brasília, Brasília, Brazil, in 2013, and the M.Sc. and Ph.D. degrees in geography from the University of Brasília, Brasília, Brazil, in 2016 and 2019, respectively.

He is currently a Geographer with the Chico Mendes Institute for Biodiversity Conservation, Brasília, Brazil, and works in projects involving remote sensing and earth sciences with the Laboratory of Space Information and Systems.

**Renato Fontes Guimarães** (Member, IEEE) received the B.Sc. degree in cartography engineering from the Universidade do Estado do Rio de Janeiro, Angra dos Reis, Brazil, in 1987, the M.Sc. degree in geophysics from the Observatório Nacional, Rio de Janeiro, Brazil, in 1991, and the Ph.D. degree in geology from the Universidade Federal do Rio de Janeiro, Rio de Janeiro, Brazil, in 2000.

He is currently a Professor with the University of Brasília, Brasília, Brazil. His main research interests include geoscience, remote sensing, geomorphology, mathematical modeling, and geoprocessing.

**Roberto Arnaldo Trancoso Gomes** (Member, IEEE) received the B.Sc., M.Sc., and Ph.D. degrees in geography from the Federal University of Rio de Janeiro, Rio de Janeiro, Brazil, 1999, 2002, and 2006, respectively.

He is currently an Associate Professor with the Geography Department, University of Brasília, Brasília, Brazil, and a Professor of the Graduate Program in Geography with the University of Brasília. He is an Editor of Revista Espaço and Geografia and a Reviewer in several scientific journals. He has experience in Geosciences and Geomorphology, mapping, process modeling, digital image processing, and artificial intelligence.

**Díbio Leandro Borges** (Senior Member, IEEE) received the B.S. degree in electrical engineering, the M.S. degree in computer science from the University of Brasília (UnB), Brasília, Brazil, in 1986 and 1991, respectively, and the Ph.D. degree in computer science from The University of Edinburgh, Edinburgh, U.K., in 1996.

He is currently an Associate Professor with the Computer Science Department, UnB. His interests in research span from computer vision, machine learning, visual perception, remote sensing, and robotics, to smart applications in precision agriculture and the environment.