Journal of Urban and Environmental Engineering, v.16, n.1, p.45-54

ISSN 1982-3932
doi: 10.4090/juee.2022.v16n1.045054

Journal of Urban and Environmental Engineering

www.journal-uee.org

# INDIVIDUAL TREE IDENTIFICATION IN URBAN AREAS FROM AERIAL IMAGES USING MASK R-CNN

Rafael T. Uematsu[1], Francisco A. Silva[1*], Leandro L. Almeida[1], Danillo R. Pereira[1], Almir O. Artero[2] and Marco A. Piteri[2]

[1]*Department of Computer Science, University of Western São Paulo (Unoeste), Presidente Prudente, São Paulo, Brazil*

[2]*Department of Mathematics and Computer Science, Faculty of Science and Technology, São Paulo State University (Unesp), Presidente Prudente, São Paulo, Brazil*

Received 16 January 2021; received in revised form 10 May 2022; accepted 10 June 2022

**Abstract:** Trees in urban centers can bring many benefits to population health. Each city must be responsible for the planning and management of urban forestation, but it is difficult to check if all places are properly wooded. With technological advances, there is the possibility of developing tools to help these tasks in a computational way. In this paper, we present a low-cost tree identification method that uses a Mask R-CNN deep neural network. The experiments performed presented a correct rate of 91.39% in the identification of the trees from aerial photographs obtained by drone.

**Keywords:** Tree identification; Tree segmentation; Aerial images; Mask R-CNN.



[*]Correspondence to: Francisco A. Silva, Tel.: +55 18 3229 1060. E-mail: chico@unoeste.br

## INTRODUCTION

Forestation is extremely important in urban centers and is responsible for numerous benefits that help in the quality of life in cities and in the physical and mental health of the population (Cechetto, *et al*., 2014). Among the benefits offered by trees are shadow for pedestrians and vehicles, reduction of noise pollution, improvement in air quality, and reduction of thermal amplitude, beyond its landscape function, which contributes to the esthetics of cities (Silva-Filho *et al*., 2002).

Lima (2016) says that the ideal places for a large concentration of trees in the cities are forests, reserves, and parks. However, they are not only concentrated in these places, being common the presence of trees distributed along the streets so that they are close to people's homes.

Due to these important factors in urban centers, the minimum amount advisable by the World Health Organization (WHO) is 12m² of green area per habitant, and the ideal is 36m², about three trees per resident (Menezes, 2016). In the municipal law of Presidente Prudente city of No. 7551/2011, it is required the planting of trees on the sidewalks of all residential and commercial properties, as well as future allotments to be implemented (Law No. 7551/2011, 2019). In addition, the municipal law No. 9297/2017 of the same city proposes the guarantee of a discount on the Urban Property Tax (UPT) for owners who maintain their sidewalks afforested, encouraging the planting and maintenance of trees (Law No. 9297/2017, 2019).

Each municipality is responsible for planning and managing its urban forestation, but there is great difficulty for public administrators to verify that all sites are properly forested. The traditional methods adopted by public administrators employ inspectors to visit all regions of the city and make a mapping of the trees, causing a lot of work time.

Automatic methods of tree detection from images and sensor data are increasingly being recommended to improve decision-making for tree management and planting, mainly due to the technological resources available for acquiring aerial and satellite (orbital) images (Lima, 2016).

There are several works in the literature for tree identification, but they use specific resources such as Light Detection and Ranging (LiDAR) that are not accessible to everyone (Zhang *et al*., 2008) (Kolb *et al*., 2015) (Mak & Hu, 2015) and Near Infrared (NIR) (Zhu *et al.*, 2016) (Mozgeris *et al.*, 2016). Komura & Muramoto (2007) used high-quality orbital images for data analysis, which are not available freely.

This work contributes to a low-cost methodology for tree identification using only digital photographic images. The acquisition of high-quality orbital images and sensor data is very expensive financially compared to, for example, obtaining aerial images using drones.

For this, we use computer vision techniques and artificial intelligence to identify trees in aerial images. To perform identification, we use a Mask R-CNN, which is a deep neural network designed to solve instance segmentation problems (explained in Section 2) in machine learning (He *et al*., 2017).

The other sections of this paper are organized as follows: Section 2 describes image segmentation; Section 3 describes Mask R-CNN with its fundamental components; Section 4 presents the Proposed Method, including the training dataset, Mask R-CNN training, and evaluation of results; Finally, Section 5 presents some conclusions and suggestions for future work.

## SEGMENTATION

One of the most difficult problems in computer vision has been image segmentation, which means dividing the image into groups of pixels based on some criteria, such as edges, colors, and textures, among others. This is different from image classification or object recognition because it is not necessary to know which are the visual concepts or objects previously (Nayak, 2018).

There is a difference between image classification, object detection, semantic segmentation, and instance segmentation. In **Fig. 1**, an image is shown for clarification regarding the differences between classification, detection, and the types of segmentation (semantic and instance).

Image classification aims to provide a set of labels to characterize the content of an input image. Object detection is based on image classification, making it possible to locate each object in an image. In segmentation, there are two types: semantic segmentation and instance segmentation (Rosebrock, 2018).

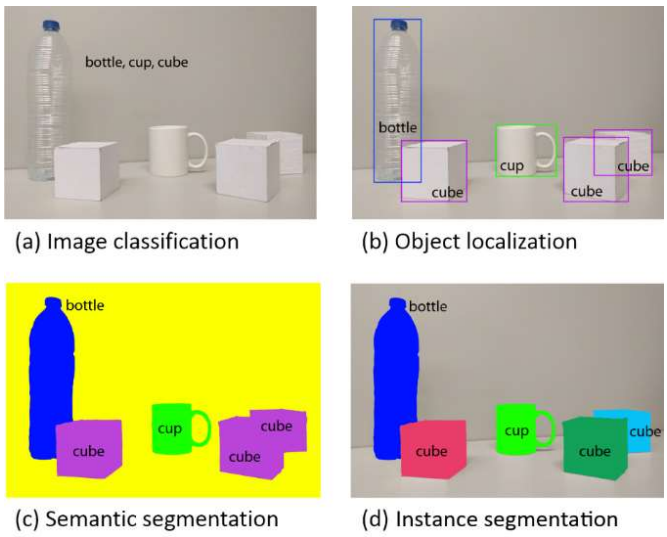



**Fig. 1** (a) Image classification. (b) Object detection. (c) Semantic segmentation. (d) Instance segmentation. Adapted from (Garcia-Garcia *et al*., 2017).

Semantic segmentation algorithms associate each pixel of the input image with class labels, including a class label for the background. Although they are able to label all objects in an image, they cannot differentiate between two or more objects of the same class, as can be seen in **Fig. 1(c)**. In this item, the three cubes present in the image have the same color, that is, it is not possible to distinguish one cube from another when they are overlapping. Instance segmentation algorithms compute a pixel mask for each object in the image, even if the objects are from the same class label. In the example in **Fig. 1(d)**, we can see that each cube has a different color, that is, this segmentation makes it possible to differentiate each cube individually in the image (Rosebrock, 2018).

## MASK R-CNN

Mask R-CNN is based on previous work of object detection using R-CNN (Girshick *et al*., 2014), Fast R-CNN (Girshick, 2015), and Faster R-CNN (Ren *et al*., 2015). Mask R-CNN architecture is an improvement over Faster R-CNN, which has two outputs for each candidate object, a class label, and an offset of a bounding box. Thus, in Mask R-CNN, a third branch is added that produces the object mask. In addition, a ROIPooling module is replaced by a ROIAlign module (He *et al*., 2017). **Fig. 2(a)** contains an architectural representation of a Faster R-CNN, and **Fig. 2(b)** is an architectural representation of a Mask R-CNN.

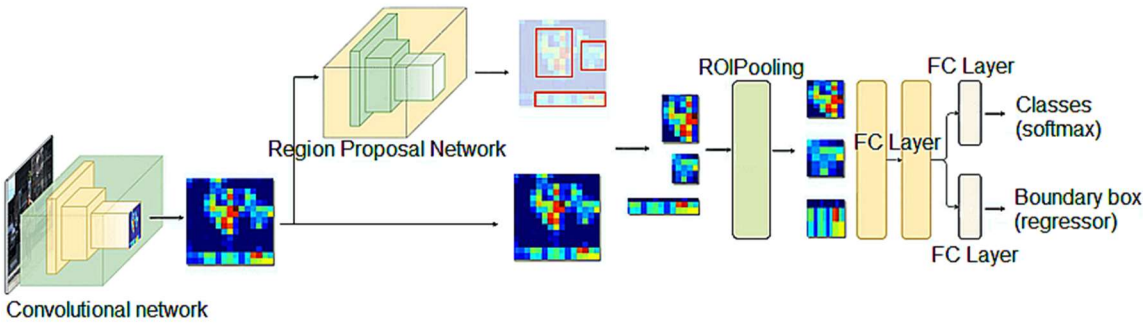
Following are the six fundamental components of a Mask R-CNN network, being: 1) Convolutional Neural Network (CNN), 2) Residual Neural Network (ResNet), 3) Feature Pyramid Network (FPN), 4) Region Proposal Network (RPN), 5) Region of Interest Alignment (ROIAlign) and 6) the output of Mask R-CNN.

## Convolutional Neural Network

A Convolutional Neural Network (CNN) is currently one of the main categories for image recognition and classification. CNNs are multilayer networks designed to recognize pixel-formed patterns without the need for complex pre-processing. CNNs are able to recognize extremely difficult patterns and are adaptable to distortions and geometric transformations (Prabhu, 2018) (Guo *et al*., 2017) (Yang & Li, 2017).

CNNs can have dozens or hundreds of layers in which each one learns to detect different characteristics of an image. These layers perform operations that change the data with the intention of learning specific features of the image. An example of CNN architecture is illustrated in **Fig. 3**, where the three most common layers are presented: convolution, ReLU, and pooling. After learning in several layers, CNN architecture changes to the classification layers, which are the fully connected neural network layers.

(a)




(b)

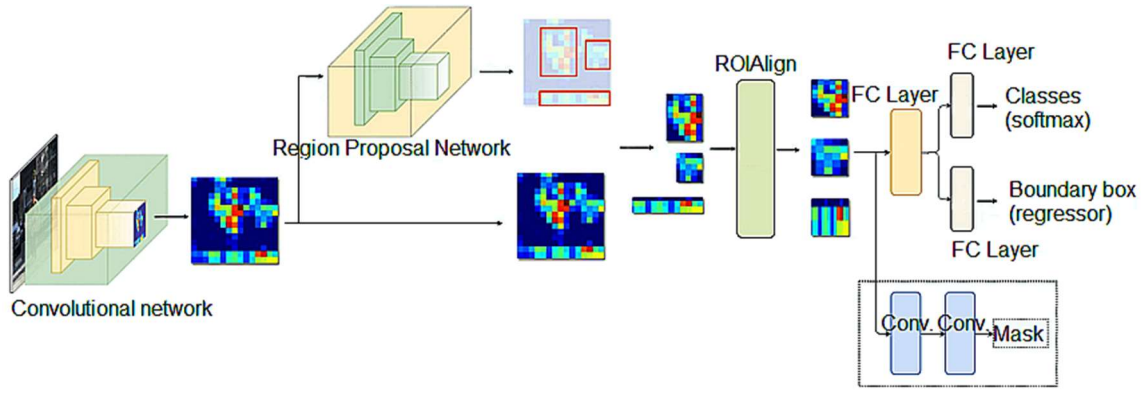



**Fig. 2** Network architectures: (a) Faster R-CNN; (b) Mask R-CNN. Adapted from Hui (2018b).

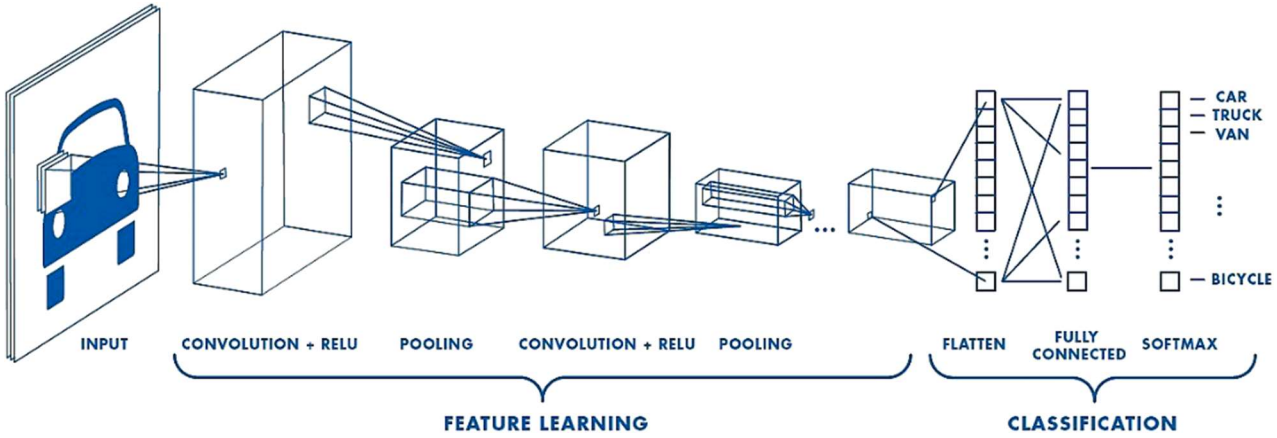



**Fig. 3** Illustration of an example of CNN architecture, adapted from Prabhu (2018).

Convolutional layers are responsible for extracting features from an input image. Convolutions consist of a small set of filters but extend to the entire depth of the input image. During the network training process, these filters are adjusted so that they are activated in the presence of relevant characteristics identified in the input volume, such as edge and color orientation. Each of these filters gives rise to a locally connected structure that runs across the entire length of the input image. The scalar product between filter values and each input volume position is an operation known as convolution (Albawi *et al*., 2017).

The resulting values after the convolution operation go through an activation function, and the most commonly used is the ReLU function (Rectified Linear Units). This operation allows for faster and more effective training by mapping negative values to zero while maintaining only positive values (Ide & Kurita, 2017). Pooling layers are responsible for reducing the resulting spatial dimension after convolutional layers, aiming to reduce the computational cost of the network and also to maintain important information from the input image. For this, values belonging to a particular region of the feature map, generated by convolutional layers, are replaced by some metrics from that region, such as maximum value (max-polling), minimum value (min-polling) or average value (avg-polling) (Araújo *et al*., 2017).

The output of the convolutional and pooling layers represents the features extracted from the input image. The purpose of the fully connected layers is to use these features to classify the image into a predetermined class. The fully connected layers are the same as a multi-layer artificial neural network (Multi-Layer Perceptron – MLP) that uses the softmax activation function on the last layer (Karn, 2016). Softmax is a function that receives a vector of values as input and produces the probabilistic distribution of the input image belonging to each of the classes in which the network was trained (Wang *et al*., 2018).

## Residual Neural Network

A residual neural network (ResNet) (He *et al*., 2016) is a network that differs from conventional CNNs by incorporating a technique to prevent data degradation at the deeper layers of the network.

A ResNet has residual blocks, each composed of convolution layers and ReLU activation function, as shown in **Fig. 4**. The number of these residual blocks defines the depth of the ResNet network. These residual blocks differ from the other convolutional layers because there is the idea of using shortcuts that jump the groups of convolutional layers (represented in Fig. 4 by the arrow). This prevents a deeper network from simply stacking layers that do nothing. ResNet architecture is a structure of four types of residual blocks that can be used to build networks of different depths.

**Table 1** presents specific ResNet models described by He *et al*. (2016). The ResNet of 18 and 34 layers uses two deep layers, while the ResNet of 50, 101, and 152 layers uses three deep layers.

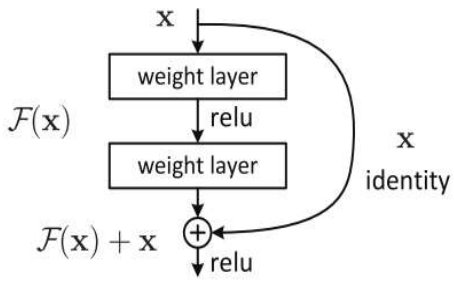



**Fig. 4** Example of a ResNet architecture residual block containing two convolution layers (weight layer) and one activation layer (ReLU) (He *et al*., 2016).

**Table 1.** ResNet architecture models (He *et al*., 2016).

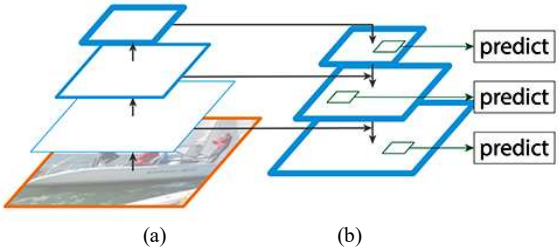
| layer name | output size | 18-layer | 34-layer | 50-layer | 101-layer | 152-layer |
|---|---|---|---|---|---|---|
| conv1 | 112×112 | 7×7, 64, stride 2 | | | | |
| conv2_x | 56×56 | 3×3 max pool, stride 2 | | | | |
| | | [3×3, 64; 3×3, 64] ×2 | [3×3, 64; 3×3, 64] ×3 | [1×1, 64; 3×3, 64; 1×1, 256] ×3 | [1×1, 64; 3×3, 64; 1×1, 256] ×3 | [1×1, 64; 3×3, 64; 1×1, 256] ×3 |
| conv3_x | 28×28 | [3×3, 128; 3×3, 128] ×2 | [3×3, 128; 3×3, 128] ×4 | [1×1, 128; 3×3, 128; 1×1, 512] ×4 | [1×1, 128; 3×3, 128; 1×1, 512] ×4 | [1×1, 128; 3×3, 128; 1×1, 512] ×8 |
| conv4_x | 14×14 | [3×3, 256; 3×3, 256] ×2 | [3×3, 256; 3×3, 256] ×6 | [1×1, 256; 3×3, 256; 1×1, 1024] ×6 | [1×1, 256; 3×3, 256; 1×1, 1024] ×23 | [1×1, 256; 3×3, 256; 1×1, 1024] ×36 |
| conv5_x | 7×7 | [3×3, 512; 3×3, 512] ×2 | [3×3, 512; 3×3, 512] ×3 | [1×1, 512; 3×3, 512; 1×1, 2048] ×3 | [1×1, 512; 3×3, 512; 1×1, 2048] ×3 | [1×1, 512; 3×3, 512; 1×1, 2048] ×3 |
| | 1×1 | average pool, 1000-d fc, softmax | | | | |
| FLOPs | | $1.8\times10^9$ | $3.6\times10^9$ | $3.8\times10^9$ | $7.6\times10^9$ | $11.3\times10^9$ |

## Feature Pyramid Network

Detecting objects at different scales is a fundamental challenge in computer vision. A widely used strategy to perform this task is to generate different scales of the same image to detect objects, but the processing is very time-consuming, and the memory demand is very high. Alternatively, feature maps are produced from the image using the convolutional and pooling layers. However, these feature maps closest to the image layer are composed of low-level structures that are not effective for accurate object detection (Hui, 2018a).

Feature Pyramid Network (FPN) (Lin *et al*., 2017) is an architecture to extract features from an image, aiming at accuracy and speed. This method generates feature maps at various scales with better-quality information to detect objects. For this, the FPN is composed of bottom-up and top-down paths, as shown in **Fig 5**. The bottom-up path is a convolutional network for extracting features from images. As the convolutional layer and pooling are applied, the spatial resolution decreases, but with the high-level structure detected, the semantic value of each layer will increase, resulting in a semantically rich layer (Lin *et al*., 2017).

FPN provides the top-down path for building different scale layers from a semantically rich layer (the top layer of the pyramid). While reconstructed layers are semantically rich, object locations are not accurate, so there are lateral connections between reconstructed layers and corresponding feature maps to help the detector predict the best locations to find an object (Lin *et al*., 2017).




(a) (b)

**Fig. 5** Feature Pyramid Network Architecture (FPN). (a) Bottom-up path. (b) Top-down path (Lin *et al*., 2017).

## Region Proposal Network

Region Proposal Network (RPN) (Ren *et al*., 2015) is an algorithm to generate a number of bounding boxes called the Region of Interest (ROI), which has a high probability of containing some object.

To generate ROIs, a small sliding window is used spatially on the feature map. At each sliding window location, a mechanism called an anchor is created, as shown in **Fig. 6**. Anchors are the central points of sliding windows, which are references to help detect objects of different proportions and scales. When the sliding window goes through the entire feature map, many ROIs are generated in a single image. In order not to maintain all ROIs, RPN ranks them with the highest probability of getting an object. Moreover, there is the possibility of more than one ROI for the same object. In this case, the ROI that most represents the object is maintained.

## ROIAlign

Applying RPN produces different-size ROIs, but it is not simple to create an efficient structure for working with feature maps with these varying sizes. Therefore, it is necessary to reduce the ROIs to the same defined size (Machine-Vision Research Group, 2018).

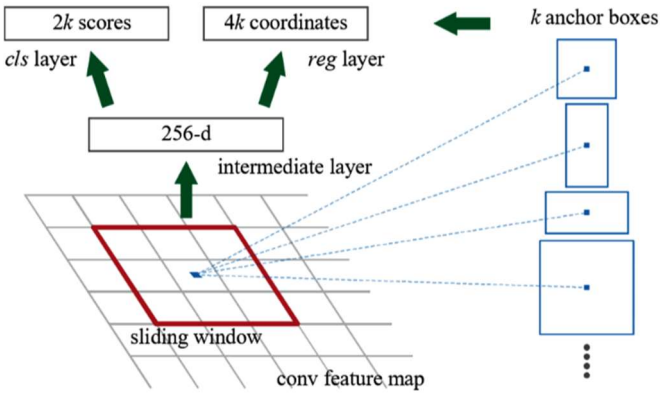



**Fig. 6** Anchor representation in the Region Proposal Network sliding window (RPN) (Ren *et al*., 2015).

In **Fig. 7(a)**, a feature map with dimension 5x5 (continuous lines) and an ROI (dashed lines) are illustrated. The limits of this ROI do not match the granularity of the feature map, so the ROI limits are rounded to match this granularity, as illustrated in **Fig. 7(b)**. Depending on the output size of the ROIs, an *m*x*m* matrix is obtained. In **Fig. 7(c)**, $m = 2$ was used, creating a 2x2 matrix. This operation causes the limits of the matrix cells not to match the granularity of the feature map. Thus, the matrix cells are divided into the granularity limits of the feature map, as shown in **Fig. 7(d)**. Finally, the max-pooling method is applied to the matrix, and ROIs of the same size are generated, as illustrated in **Fig. 7(e)**.

This ROIPooling method is used in Faster R-CNN, which works for object detection cases, but fails in segmentation cases because there are steps that require the offset in the ROIs pixels, as shown in **Fig. 7(a)** and **Fig. 7(c)**. For these reasons, ROIPooling has been replaced by the ROIAling method.

In **Fig. 8**, the same 5x5 feature map and the ROI of **Fig. 7(a)** are illustrated, but in the ROIAlign technique, no offset operation is performed; instead, the coordinates at the ends of the ROI are labeled, and the *m*x*m* matrix is obtained as output, where $m = 2$.

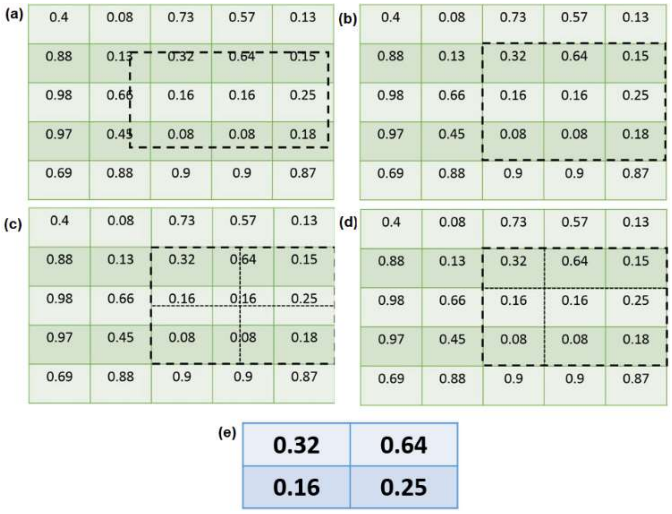



**Fig. 7** (a) 5x5 feature map and an ROI. (b) ROI offset to match the feature map granularity. (c) 2x2 matrix under ROI. (d) Matrix cells corresponding to feature map granularity. (e) Result of applying the max-pooling method (Machine-Vision Research Group, 2018).

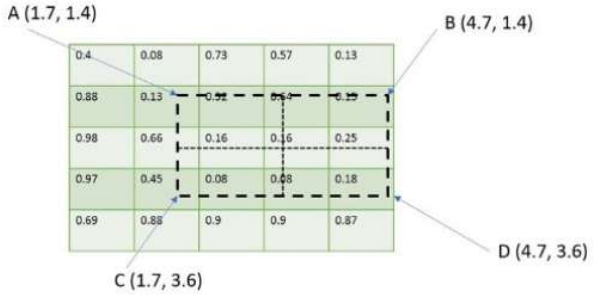



**Fig. 8** 5×5 feature map and an ROI (Machine-Vision Research Group, 2018).

In each cell of this matrix (**Fig. 8**), sampling points are created, these points can be calculated by **Eq. (1)** and **Eq. (2)**.

$$x = x_{low} + (i + 0.5) * \frac{x_{high} - x_{low}}{n} \quad (1)$$

$$y = y_{low} + (i + 0.5) * \frac{y_{high} - y_{low}}{n} \quad (2)$$

ROIAlign calculates the value of each sampling point by bilinear interpolation in relation to the grid points closest to the feature map, as shown in **Fig. 9**. Thus, different values are obtained for each sampling point, and the avg-pooling method is applied to acquire a final output of equal dimensions for all ROIs.

### Mask R-CNN output

In Mask R-CNN output, the ROIs obtained after applying the ROIAlign technique are used to obtain the prediction of the classes, bounding boxes, and masks.

For the class prediction and bounding boxes, the ROIs are remodeled for the fully connected layer's input. For this, a vector of the same size is generated for each ROI, containing two branches, each with a fully connected layer, to predict the object class and another for regression values of bounding boxes (He *et al*., 2017).

In order to extract features from the masks, a fully convolutional network (FCN) is used, which is built only by convolution layers, ReLU, and pooling, which are applied in ROIs to generate low-resolution masks, and thus examine the existence of the object at this level of granularity. Finally, an upsampling operation is used to resize the ROI to the original dimension (He *et al*., 2017) (Shelhalmer *et al*., 2017).

## PROPOSED METHOD

This section describes how the proposed method works, being divided into five steps: training dataset, pre-processing, used model, training and results evaluation.


**Fig. 9** Bilinear interpolation at the sampling point (Machine-Vision Research Group, 2018).

### Training Dataset

The training dataset used in this work consists of aerial images of urban areas, which were acquired from the senseFly website (senseFly Parrot Group, 2019). Drones with high-quality cameras have captured these images in various locations in Switzerland. In this work, 200 images from various regions containing a large number of trees were used to favor the training of Mask R-CNN.

Beyond the images obtained from the senseFly website, a weight file already trained with the MS COCO (2018) dataset was used so that it was not necessary to train a model from the beginning. This file was used because training a deep learning model requires a very large dataset. Although this dataset does not contain the tree class, it does contain many other classes, so trained weights represent characteristics commonly found in many images.

### Pre-processing

The images obtained on the senseFly website have varying resolutions with a maximum of 6000x4000 pixels. In order to build the training image dataset, the original images were cropped to a lower resolution (1024x1024 pixels).

For each original image (200 images), an average of three new images were obtained, producing 600 images for training. The reason for using smaller images than the original ones in training was to reduce the computational cost.

In order to use a Mask R-CNN, it was necessary to annotate the objects (trees) in the images. This work was done manually using the VIA software (VGG Image Annotator, 2019). With this, masks were created using polygon points around each tree (object of interest) in the images, and a file was generated in JSON format.

### Used Model

There are several convolutional neural network models that have as their main objective the classification, detection, and segmentation of objects. The big difference between the models is related to the number of convolutional layers present, the number of hidden neurons used in the fully connected layers, and the size of the filters used in the convolutional layers. These parameters directly affect the accuracy and computational cost of the model.

In the implementation of this work was used a 101 layer ResNet-based Mask R-CNN and Feature Pyramid Network (FPN). This model generates bounding boxes, classes, and segmentation masks for each instance of an object in the output.

### Training

TensorFlow (2018) Python library was used for the training. This library is intended to facilitate the development of applications that use high-performance numerical computing. The training process was performed on a computer with a 3.6 GHz AMD Ryzen 5 1600X processor with 8GB of RAM.

The image dataset was divided for training and testing, 90% of the images were used in training, and 10% of the images were used in the test.

Furthermore, some parameters necessary for the training were defined. These parameters are fixed and do not change during the training, they are fundamental for neural network efficiency. The parameters used in training were the learning rate: 0.001; the number of training stages per epoch: 1000; the number of validation stages per epoch: 50; and the number of epochs: 18.

The training time was approximately two days based on 18 defined epochs and computer CPU utilization, which takes longer compared to using a GPU.

It is possible to observe the progress of training by viewing the graphs generated by the Tensorboard, which is a TensorFlow tool to simplify the visualization of the data generated during training. **Fig. 10** contains a representation of the error generated by the network during the 18 training epochs.

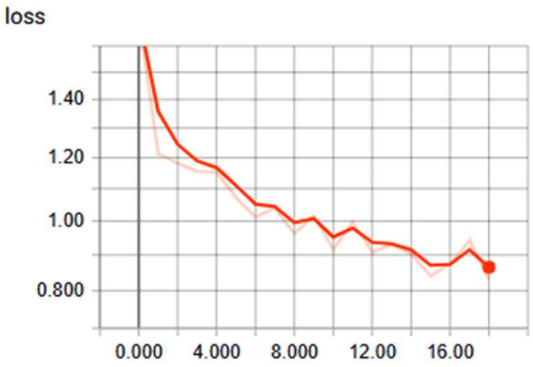



**Fig. 10** Tensorboard generated graph representing error during training.

### Results Evaluation

In order to perform the final evaluation of the network, 56 images were used for validation, and these images were not included in the training dataset. Some examples of the results obtained can be seen in **Fig. 11**.

**Fig. 11** Image samples with result applying the Mask R-CNN modeled in this work.

The proposed method achieved very satisfactory results in the process of identifying the trees, as it was able to identify 223 of the 244 trees in the 56 images used, thus obtaining a hit rate of 91.39%, with an error of only 8.61%. However, Mask R-CNN identified 52 objects as trees, as shown in **Fig. 12**. Although an identification error occurred in the image of Fig. 12, this error is quite justifiable because the tree shadow has the exact shape of a tree, while the other various objects present in the image are properly distinguished from a tree.

**Fig. 12** Example of a tree identification error. Mask R-CNN identifies the shadow of a tree as a tree (blue contour).

The factors that make tree identification difficult are related to the amount of information in the image, such as shadows from tall buildings and shrubs, depending on the angle of the sun, and the variety of tree species.

## CONCLUSIONS

The results of this work show that the methodology developed presents a good performance, with a hit rate of 91.39% in the identification of trees. However, due to a large amount of information present in the images, some identification errors occurred.

In order to improve the results obtained and thus be able to apply the methodology developed in images from other regions of the world, it is suggested, in future works, to add new images to the training dataset to represent scenarios that the data set used did not address. In addition to improving the training dataset, it is possible to refine the training parameters, seeking to find the best parameter values that, when applied, can increase the hit rate in tree identification.

## REFERENCES

Albawi, S.; Mohammed, T.A. & Al-Zawi, S. (2017) Understanding of a convolutional neural network, In: *IEEE International Conference of Engineering and Technology* (ICET), Antalya, Turkey.

Araújo, F.H.D.; Carneiro, A.C.; Silva, R.V.; Medeiros, F.N.S. & Ushizima, D.M. (2017) Redes Neurais Convolucionais com

Tensorflow: Teoria e Prática, In: *III Escola Regional de Informática do Piauí*. Anais – ERI 2017, **1**(1), Piauí, 382–406.

Cechetto, C.T.; Christimann, S.S. & Oliveira, T.D. (2014) Arborização urbana: importância e benefícios no planejamento ambiental das cidades, In: *XVI Seminário Internacional de Educação do Mercossul*, Unicruz.

COCO – Common Objects in Context (2018). Available in: http://cocodataset.org/#home. Accessed on 2018 September 25.

Garcia-Garcia, A.; Orts-Escolano, S.; Oprea, S.O.; Villena-Martinez, V. & Garcia-Rodriguez, J. (2017) A Review on Deep Learning Techniques Applied to Semantic Segmentation, Arxiv Prepr. http://arxiv.org/abs/1704.06857.

Girshick, R.; Donahue, J.; Darrell, T. & Malik, J. (2014) Rich feature hierarchies for accurate object detection and semantic segmentation, In: *IEEE Conference on Computer Vision and Pattern Recognition, Columbus*, OH, USA, 580–587.

Girshick, R. (2015) Fast R-CNN, In: *Proceedings of the 2015 IEEE International Conference on Computer Vision* (ICCV), 2440–2448.

Guo, T.; Dong, J.; Li, H. & Gao, Y. (2017) Simple convolutional neural network on image classification, In: *IEEE 2nd International Conference on Big Data Analysis* (ICBDA), Beijing, China, 721–724.

He, K.; Zhang, X.; Ren, S. & Sun, J. (2016) Deep Residual Learning for Image Recognition, In: *IEEE Conference on Computer Vision and Pattern Recognition* (CVPR), **1**, Las Vegas, NV, USA, 770–778.

He, K.; Gkioxari, G.; Dollár, P. & Girshick, R. (2017) Mask R-CNN, In: IEEE International Conference on Computer Vision (ICCV), Venice, Italy, 2980–2988.

Hui, J. (2018a) Understanding Feature Pyramid Networks for object detection (FPN). Available in: https://medium.com/@jonathan_hui/understanding-feature-pyramid-networks-for-object-detection-fpn-45b227b9106c. Accessed on 2018 October 6.

Hui, J. (2018b) Image segmentation with Mask R-CNN, 2018. Available in: https://medium.com/@jonathan_hui/image-segmentation-with-mask-r-cnn-ebe6d793272. Accessed on 2018 October 17.

Ide, H. & Kurita, T. (2017) Improvement of learning for CNN with ReLU activation by sparse regularization, In: *IEEE International Joint Conference on Neural Networks* (IJCNN), Anchorage, AK, USA, 2684–2691.

Karn, U. (2016) An Intuitive Explanation of Convolutional Neural Networks. Available in: https://ujjwalkarn.me/2016/08/11/intuitive-explanation-convnets. Accessed on 2018 November 15.

Kolb, A.; Meaclem, C.; Chen, X.; Parker, R.; Gutschmidt, S. & Milne, B. (2015) Tree trunk detection system using LIDAR for a semi-autonomous tree felling robot, In: *IEEE 10th Conference on Industrial Electronics and Applications* (ICIEA), Auckland, New Zealand, 84–89.

Komura, R. & Muramoto, K. (2007) Classification of forest stand considering shapes and size of tree crown calculated from high spatial resolution satellite image, In: *IEEE International Geoscience and Remote Sensing Symposium*, Barcelona, Spain, 4356–4359.

Lei nº 7551/2011. Câmara municipal de Presidente Prudente. Available in: http://www.presidenteprudente.sp.gov.br/site/Documento?cod=18512. Accessed on 2019 April 15.

Lei nº 9297/2017. Câmara municipal de Presidente Prudente. Available in: http://www.presidenteprudente.sp.gov.br/site/Documento.do?cod=37306. Accessed on 2019 April 15.

Lima, H.G.F. (2016) Identificação e Estimativa da Altura de Arvores em Imagens de Satélite e do Google Street View, master thesis, Instituto de Informática, Programa de Pós-graduação em Ciência da Computação.

Lin, T.; Dollár, P.; Girshick, R.; He, K.; Hariharan, B. & Belongie, S. (2017) Feature Pyramid Networks for Object Detection, In: *IEEE Conference on Computer Vision and Pattern Recognition* (CVPR), Honolulu, HI, USA, 936–944.

Machine-Vision Research Group (2018). Mask R-CNN Unmasked. Available in: https://medium.com/@fractaldle/mask-r-cnn-unmasked-c029aa2f1296. Accessed on 2018 October 4.

Mak, H. & Hu, B. (2015) Characterization of tree structures from mobile LiDAR data for the identification of Ash trees, In: *IEEE International Geoscience and Remote Sensing Symposium* (IGARSS), Milan, Italy, 5371–5374.

Menezes, F. Z. (2016) Gazeta do Povo. Uma árvore por habitante, a recomendação minima da OMS para as cidades. Available in: https://www.gazetadopovo.com.br/vida-e-cidadania/futuro-das-cidades/uma-arvore-por-habitante-a-recomendacao-minima-da-oms-para-as-cidades-622ch9afm4rimh3ol1w9j8ikn/. Accessed on 2019 April 15.

Mozgeris, G.; Gabal, S.; Jonikavičius, D.; Straigytė, L.; Ouerghemmi, W. & Juodkienė, V. (2016) Hyperspectral and color-infrared imaging from ultralight aircraft: Potential to recogniza tree species in urban environments, In: *IEEE 8th Workshop on Hyperspectral Image and Signal Processing: Evolution in Remote Sensing* (WHISPERS), Los Angeles, CA, USA.

Nayak, S. (2018) Deep learning based Object Detection and Instance Segmentation using Mask R-CNN in OpenCV (Python / C++), 2018. Available in: https://www.learnopencv.com/deep-learning-based-object-detection-and-instance-segmentation-using-mask-r-cnn-in-opencv-python-c/. Accessed on 2018 December 14.

Prabhu, R. (2018) Understanding of Convolutional Neural Network (CNN) – Deep Learning. Available in: https://medium.com/@RaghavPrabhu/understanding-of-convolutional-neural-network-cnn-deep-learning-99760835f148. Accessed on 2018 December 5.

Ren, S.; He, K.; Girshick, R. & Sun, J. (2015) Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks, In: *Proceefings of the 28th International Conference on Neural Information Processing Systems*, **1**, 91–99.

Rosebrock, A. (2018) Mask R-CNN with OpenCV. Available in: https://www.pyimagesearch.com/2018/11/19/mask-r-cnn-with-opencv/. Accessed on 2018 November 21.

senseFly Parrot Group (2019). Available in: https://www.sensefly.com/education/datasets/. Accessed on 2019 January 8.

Shelhalmer, E; Long, J.; & Darrell, T. (2017) Fully Convolutional Networks for Semantic Segmentation, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **39**(4), 640–651.

Silva-Filho, D.F.; Pizetta, P.U.C.; Almeida, J.B.S.A.; Pivetta, K.F.L. & Ferraudo, A.S. (2002) Banco de dados relacional para cadastro, avaliação e manejo da arborização em vias públicas. *Revista Árvore*, Viçosa, **26**(5), 629–642.

TensorFlow (2018). Available in: https://www.tensorflow.org. Accessed on 2018 September 23.

VGG Image Annotator (2019). Available in: http://www.robots.ox.ac.uk/~vgg/software/via/via-1.0.6.html. Accessed on 2019 January 12.

Wang, F.; Cheng, J.; Liu, W. & Liu, H. (2018) Additive margin softmax for face verification, In: *IEEE Signal Processing Letters*, **25**, 926–930.

Yang, J. & Li, J. (2017) Application of deep convolution neural network, In: *IEEE 14th International Computer Conference on Wavelet Active Media Technology and Information Processing* (ICCWAMTIP), Chengdu, China, 229–232.

Zhang, W.; Hu, B.; Jing, L.; Woods, M.E. & Courville, P. (2008) Automatic Forest Species Classification using Combined LIDAR

Data and Optical Imagery, In: *IEEE International Geoscience and Remote Sensing Symposium* (IGARSS), Boston, MA, USA, 134-137.

Zhu, C.; Gong, H.; Li, Z. & Yu, C. (2016) Application of High Dimensional Feature Grouping Method in Near-Infrared Spectra of Identification of Tobacco Growing Areas, In: *IEEE 3rd International Conference on Information Science and Control Engineering* (ICISCE), Beijing, China, 230–234.