Deep learning for mapping rainwater drainage networks using Remote Sensing Data

Júlia Potratz[1], Cristian Enrique M. Villalobos[1], Smith W.A. Canchumuni[1], Marco Aurélio C. Pacheco[1]

[1]*Dept. of Electrical Engineering, Pontifical Catholic University of Rio de Janeiro*
R. Marquês de São Vicente, 225, 22541-041, Rio de Janeiro, Brazil
jupotratz@gmail.com, cmunozv@uni.pe, saraucoc@uni.pe, marco@ele.puc-rio.br

Abstract. Mapping rainwater drainage networks are traditionally lead from image visual interpretation. With the emergence of automatic extraction algorithms, remote sensing areas began to modernize. To improve the spatial resolution image, several works have addressed strategies to generate drainage networks, however, mapping accuracy and computational resources needed to process this information are still a problem. This work explores the potential of applying different Deep Learning approaches to the process of extraction information in rainwater drainage networks through digital elevation models obtained by the Shuttle Radar Topography Mission (SRTM). To perform this, we evaluated three different architectures: U-Net, DeepLab, and Cycle GANs (Generative Adversarial Networks). The results show that the average intersection over union (IoU) to determine the drainage networks proved superior in relation to a decision tree used as baseline that proved unsatisfactory to solve the proposed problem. However, the proposed UNet, DeepLab + FCN and Cycle GAN networks showed averages equal to 93.89% 81.13% and 92.42% respectively. These results indicate that it is possible to perform the geoprocessing of large-scale images almost in real time, making it an excellent resource to contribute to the mapping of drainage networks.

Keywords: Rainwater Drainage Networks, Semantic Segmentation, Deep Learning

1 Introduction

The Earth's surface is constantly changing through endogenous and/or exogenous processes that shape the relief. Endogenous processes are those related to tectono-magmatic processes acting on the Earth's surface below (crust and mantle). Exogenous processes are those that act on the Earth's surface, resulting from mobile agents, such as rivers, winds, tidal movements, temperature variations, and glacier movement. Regarding exogenous agents, rivers are the main modelers of terrestrial relief.

Depending on the position of the river course, different geological and geomorphological processes can be observed. Monitoring and following these processes becomes fundamental for decision-making and behavioral analysis of the land surface and the course of its rainfall networks.

With the advent of Geographic Information Systems (GIS), Digital Elevation Models (DEMs) have played an important role in acquiring data sets such as flow direction, flow accumulation, and drainage area contribution from a point, providing basic hydrological information of a watershed for runoff analysis and sediment transport studies [1, 2]. Automated extraction of datasets from a DEM was pioneered by [3] with the introduction of a deterministic eight-direction single flow (D8) algorithm, which is implemented in many GIS programs (e.g. ArcGIS®, IDRISI®, GRASS® and PCRaster®). Later, more sophisticated but less well-known algorithms were developed and are currently available, including Multiple Flow Direction (MFD) algorithms [4–7].

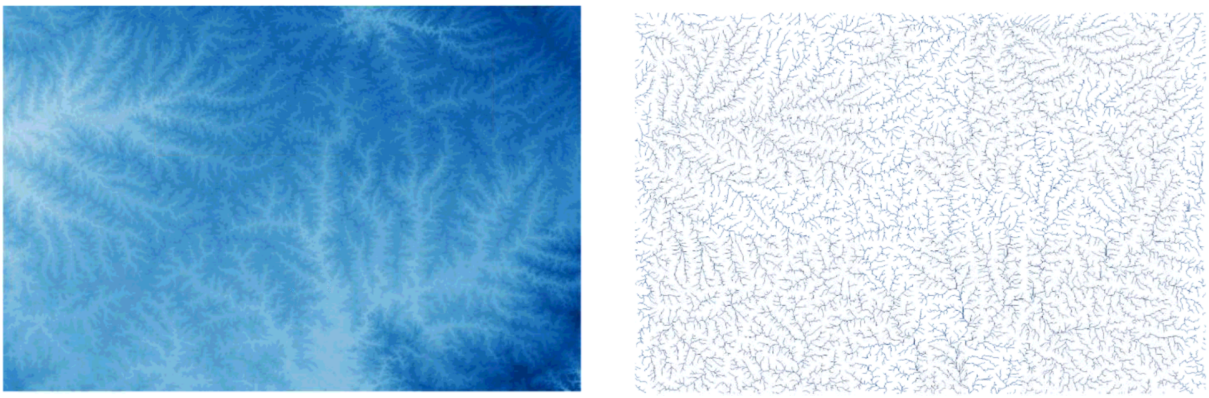
In order to extract drainage networks closer to reals is necessary to consider variables such as drainage data, flow algorithms, operation parameters, and the geomorphometric characteristics of the watersheds.

Currently, geoprocessing software is used to extracting information about drainage networks identification. However, depending on the computational capacity and characteristics of the input data, the process may become unfeasible if a very large amount of data need to be computed in a short period of time. In addition, the use of software demands a high financial cost, which can turn the business unviable [8].

A practical example of the processing carried out by ArcGIS® is shown in the figure 1. The figure 1(a) was obtained from SRTM data provided by the USGS [1]. The figure 1(b), with dimensions equal to 1800×1200 pixels, was generated using ArcGIS and its processing took about 30 minutes on a computer Core i7 with 8 GB of RAM.

Technological advances, especially with the advent of GPUS's have allowed the use of deep neural networks for machine learning, making it possible to perform previously unimaginable tasks. This new way of learning has shown interesting

[1]Public data available in `https://www.embrapa.br/satelites-de-monitoramento/missoes/srtm`



(a) SRTM image in blue scale.

(b) Reference Image generated by ArcGIS® in blue scale.

Figure 1. SRTM and reference image sample.

performance in several computer vision tasks [9].

Classification techniques, clustering, semantic segmentation, object detection, among others can be implemented using different types of neural network architectures. Previous works indicate that semantic segmentation techniques through deep learning are effective in remote sensing areas [10–12]. Thus, this approach shows a new scope for drainage network detection making use of remote sensing images. We propose two fully convolutional neural networks architectures for stormwater drainage network detection using remote sensing images. In addition, as a comparative measure, a simple pixel-by-pixel classification method using image patches as attributes for a Random Forest classification was used to become the baseline of this work.

2 Fundamentals

Computer vision focuses on develops theory and technology for building artificial systems that extract information from images or any multidimensional data. From an engineering point of view, it aims to automate tasks that usually would be done by the human visual systems. In the last years, the increase of computational power has allowed the hegemony of deep learning methods in computer vision applications, especially for image segmentation and object detection.

Semantic segmentation is a challenging task in computer vision that consists of partitioning an image into smaller fractions, called pixels, and labeling these pixels from a set of classes. In this task, the challenges faced are linked to the characteristics of the images such as appearance, texture, illumination, among others [13]. Despite the great advances in deep learning models in recent years, these problems have not yet been solved [14, 15].

Computational limitations to perform convolution operations while maintaining image dimensions have driven the emergence of architectures "encoder-decoder" that perform an undersampling (*downsamplig*) and oversampling (*upsampling*) processes. In 2017, was presented one of the first encoder-decoder architecture, known as Segnet [16].

Then [17] introduces U-Net that increases the performance of semantic segmentation. The under and oversampling operations in previous encoder-decoder architectures usually lose the spatial features during the forward propagation. To solve this problem was introduced "skip connection", which aim to preserve details that eventually have been lost during the encoding process [17]. The connections are performed through the concatenation of the corresponding layers between the encoder and the decoder.

Different implementations of fully convolutional networks have been proposed, seeking to improve the segmentation, especially the process of preserve spatial information. An important improvement in convolutional neural networks was the introduction of dilated convolutions (*atrous convolution*), which use higher spatial dimensions filters in which zeros are inserted interspersed along each dimension [18]. In addition, DeepLab integrated Global Polling operation, which preserves the main features of the images [19]. The figure 2 presents the illustrative schema of DeepLab-V3 architecture [20].

In 2014, [21] introduced a new approach in deep machine learning: Generative Adversarial Networks (GANs). The proposed architecture and the concepts behind it were considered one of the most interesting ideas in the last 10 years in the field of machine learning. GANs use the concept of a two-player game in the training process. Basically, it is a game between a generator and a discriminator who try to outdo each other by performing competitive training. The generator aims to generate samples from a random distribution while the discriminator distinguishes whether or not the generated data is real or fake.

GANs became popular in different applications such as image, video, and text generation. Since then, new concepts were introduced that evolved this architecture, such as the insertion of conditions (conditional GANs) to generate new images [22] and also applications aiming to generate images from one domain to another (cyclic GANs [23]).

Cyclic GANs (Cycle GANs) are used in problems where paired data doesn't exist. In the case of images, for each image in the $\mathbf{X}$ domain, it is desired to obtain a corresponding version in the $\mathbf{Y}$ domain. In this way, The network proposes to learn both the mapping: $\mathbf{G} : \mathbf{X} \rightarrow \mathbf{Y}$, as well as the inverse mapping $\mathbf{F} : \mathbf{Y} \rightarrow \mathbf{X}$. For this, a cyclic consistency loss function is introduced. The total loss function is given by the equation 1.

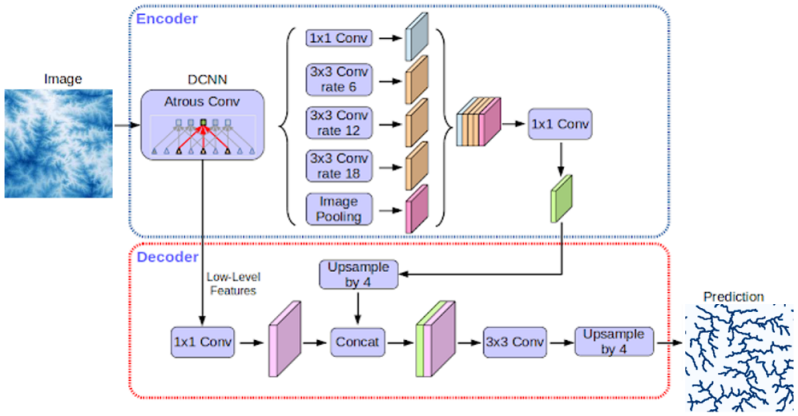


Figure 2. Schematic of the DeepLab architecture implemented.

$$\mathcal{L}(\mathbf{G},\mathbf{F},\mathbf{D_X},\mathbf{D_Y}) = \lambda_{GAN}\mathcal{L}_{GAN}(\mathbf{G},\mathbf{D_Y},\mathbf{X},\mathbf{Y}) + \lambda_{GAN}\mathcal{L}_{GAN}(\mathbf{F},\mathbf{D_X},\mathbf{Y},\mathbf{X}) + \lambda_{cyc}\mathcal{L}_{cyc}(\mathbf{G},\mathbf{F}) + \lambda_{ident}(\mathbf{G},\mathbf{F}) \quad (1)$$

Where each term denotes, respectively, the loss of representation from domain $\mathbf{X}$ to domain $\mathbf{Y}$, analogously to the loss of representation from domain $\mathbf{Y}$ to domain $\mathbf{X}$, the loss due to cyclic consistency ($\mathbf{F_Y}(\mathbf{G}(\mathbf{X})) \approx \mathbf{X}$ - and vice versa)) and loss of identity. figure 3 shows a schematic of the training process of a cyclic GAN network.

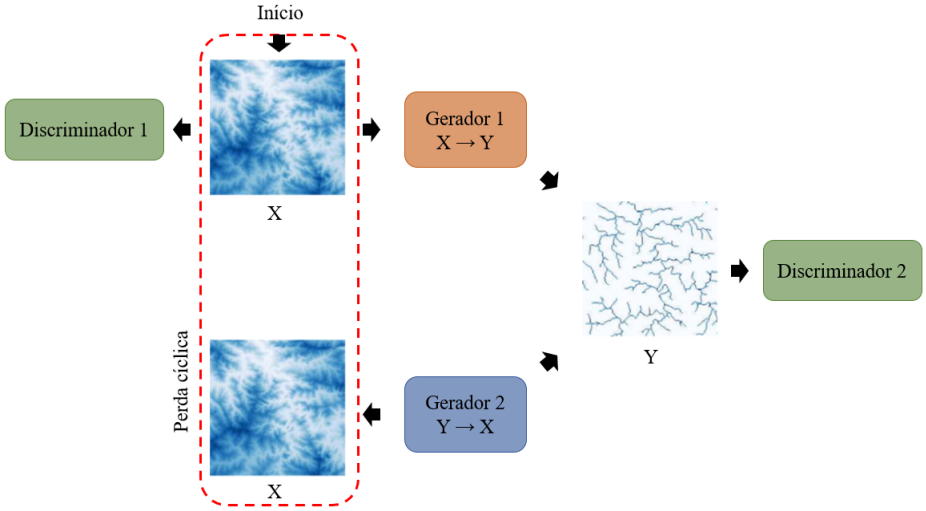


Figure 3. Schematic of the Cycle GAN architecture implemented.

2.1 Evaluation Metric

Intersection over union (IoU)

In image segmentation, the main goal is based on labeling each pixel in an image according to a specific class. IoU is a commonly used evaluation metrics to assess the performance of segmentation methods based on the labels of the reference image and the labels assigned by the network [24]. In IoU the segmented image is compared with the reference. The corresponding labels, i.e. the intersection between the two images is known as true positive. The area of the reference that was not detected by the algorithm is known as a false negative. Whereas the area of the image delivered by the model that does not match the reference is known as false positive. The intersection over the union is calculated by the ratio between the intersection of the reference image and the image delivered by the proposed model over the union between these two images.

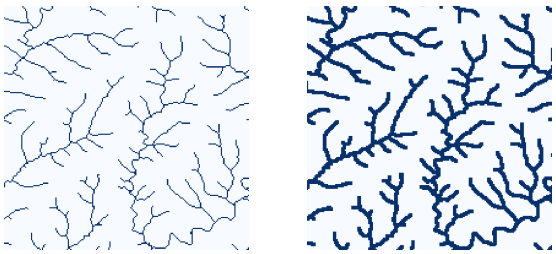
3 Methodology

3.1 Database

The database used in this work relies on 13 remote sensing images - representing digital elevation models - of dimensions 1800×1200 pixels, obtained by the SRTM (Shuttle Radar Topography Mission) mission. The images are public and available on the website of the National Institute for Space Research (INPE[2]). Each image were processed using ArcGIS® software to obtain the storm water drainage network. The image was divided into patches randomly on dimensions 128×128 pixels. Obtaining 50000 examples (patches) for each image. Although they are distinct examples, many of these patches overlap.

For training, three images were selected, for a total of 150000 patches (3×50000). The examples were normalized (mean value equal to 0 and a standard deviation equal to 1) and the labels were obtained through geoprocessing. These images were annotated with the class drainage network and class non-drainage network. The "non-drainage network", which represents the white region in the figure 1(b), occupies approximately 94% of the image, being the dominant class.

Due to the characteristics of the architecture of the neural network known as Deeplab, it was necessary to perform an initial pre-processing on the data to minimize the imbalance of classes, and additional pre-processing was performed on the label of the image. The regions where there was a stormwater drainage network (regions in blue in the figure 1(b)) were dilated, considering a radius equal to three pixels. The dilation process adopted is exemplified in figure 4, where increases the presence of class "drainage network" in the samples from 6% to approximately 21%. For other architectures, the database did not undergo this additional preprocessing.



Real label Dilated label

Figure 4. Example of a pixel dilation process considering a neighborhood equal to one. In this process, all neighboring pixels receive the same information from the central pixel, thus expanding the object within the image.

In addition to the 50000 patches, the images used for testing also underwent another preprocessing. Each image was fractionated into reduced 128×128 pixel images such that there is no overlap between the images. The original image was padding (adding zeros on the edges) to multiple dimensions of 128. Thus, 150 samples were extracted from each image. To evaluate the performance of the model to predict drainage networks, the prediction was compared with the drainage networks obtained by the geoprocessing software.

3.2 Implemented architectures

Two modified architectures for image segmentation were implemented: U-net and DeepLab. In addition, a cyclic generative adversarial network (Cycle GAN) was implemented.

For the U-Net architecture, dropout layers were added between convolutional layers of each convolutional block. The dropout layer introduces noise throughout training, and its rate was varied across the encoder and decoder layers of the network. In addition, all the convolutional layer's weights initialize with a normal distribution. For this method, the weights are initialized with the size of the previous layer, which helps to achieve a global minimum of the loss function faster and more efficiently. The weights are still random but differ in range depending on the size of the previous layer of neurons. This provides a controlled initialization, so the gradient descent becomes faster and more efficient. Finally, the batch normalization technique was not used throughout training, so the *BatchNormalization* layers were not added to the network architecture.

For DeepLabV3 architecture, represented in figure 2, the pre-trained network **Xception** was used as a backbone, which integrates depth-separable convolutions into the network architecture [25]. Despite being considered the state of the art of semantic segmentation, DeepLab presented difficulty for stormwater drainage networks segmentation. Therefore, the drainage network mask was dilated for the training with DeepLabV3. A fully convolutional network (FCN) was built and trained to perform the erosion process in the images. This network was used because the normal erosion process promoted discontinuities in the drainage networks that were being predicted by Deeplab. The FNC was built with 4 convolutional layers having between them Batch Normalization layers with an activation layer with the ReLu function. In addition, linear activation functions were used for the first three convolutional layers and the sigmoid activation function for the last convolutional layer.

Finally, the Cycle GAN performs the mapping from the remote sensing images to the drainage network domain. The training process is illustrated in figure 3. In this architecture, convolutional blocks based on ResNet were used. The configurations of the generators and discriminators are presented in the figure 5.

[2]Publicly available data and `https://www.embrapa.br/satelites-de-monitoramento/missoes/srtm`

(a) Generator Architecture

(b) Discriminator Architecture

Figure 5. Schematic illustration of the architecture of the Generators (a) and Discriminators (b) of Cycle GAN.

The formation of a ConvBlocks is composed of a convolutional layer, batch normalization layer, and an activation layer. In the formation of a ConvTransposeBlock, we have a convolutional layer, instance normalization layer, and an activation layer. In addition, ResnetBlock has a convolutional layer and two blocks of instance normalization layer and activation layer.

3.3 Experimental Setup

All experiments were performed using an NVIDIA V100 GPU 32GB. The thirteen images were divided as follows: three were used for training and validation (150000 examples) that were split into 75% and 25% for each set, respectively. The remaining ten images were used for testing. In addition, the parameter optimization process was performed on top of the training set only, to avoid overfitting problems.

All the model implementations were performed using Keras. The DeepLab and U-Net models were trained using a backpropagation algorithm with ADAM optimizer and *binary cross-entropy* loss function represented by the equation 2 where y_i is the label of the i-th pixel (1 for pixels representing drainage networks and 0 for non-representative pixels) and $p(y_i)$ is the probability that the i-th pixel is from a region with a drainage network. Whereas for Cycle GAN the lost function used is represented by the equation 2. In addition, the weights initialization for all convolutional layers were using a normal distribution with a standard deviation equal to 0.02.

$$\mathcal{L} = -\frac{1}{N}\sum_{i=1}^{N} y_i \cdot \log(p(y_i)) + (1 - y_i) \cdot \log(1 - p(y_i)) \tag{2}$$

For the training process, a learning rate decay was used with 0.001 and an early stopping with 15 epochs was used to controlling the end train condition.

4 Results and Discussions

The table 1 summarizes the performance metrics obtained by training the proposed models. The table shows the mean intersection over union (IoU) matrics for ten image test data.

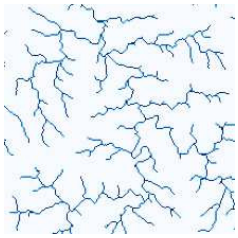
	Random Forest	U-Net	DeepLab (Dilated label)	Cycle GAN
Mean	54.5%	93.89%	81.13%	92.43%

Table 1. Mean intersection over union (IoU) metrics for ten test image (not presented to the net during the training process).

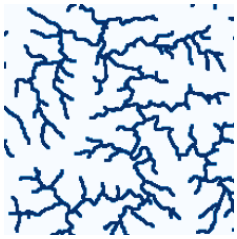
To visually evaluate the performance to segment drainage networks using the proposed architectures, fractions of one of the test images are used and are shown in figure 6. The images were represented in blue scale by convention. Table 1 and figure 7 notices an acceptable performance on U-Net architecture for the segmentation of drainage networks, having an mean IoU of 93.89%. The presence of "*skip connections*" in the U-Net architecture, which connects the encoder layers with the corresponding decoder layers, has the ability to preserve important details of the remotely sensed image. Since drainage networks are very compact, specific features of the input image are important in defining the location of these networks. In our experiments, the insertion of dropout layers inside the architecture improves the model generalization, avoiding overfitting during the training process.

For DeepLab architecture, tests were initially performed using the original labels, i.e., without any dilatation transformation. However, this approach was not successful. This fact can be explained by the DeepLab V3 use a disproportionate oversampling operation in the network decoder, which makes the network unable to correctly map the drainage networks due to their characteristics. The drainage networks represent small portions of the full image, approximately 5%. Moreover, these networks are not concentrated in a small portion of the image, they are scattered forming small objects. The DeepLab V3

(a) Reference (remote sensing image)



(b) Real drainage networks



(c) Dilated real drainage networks

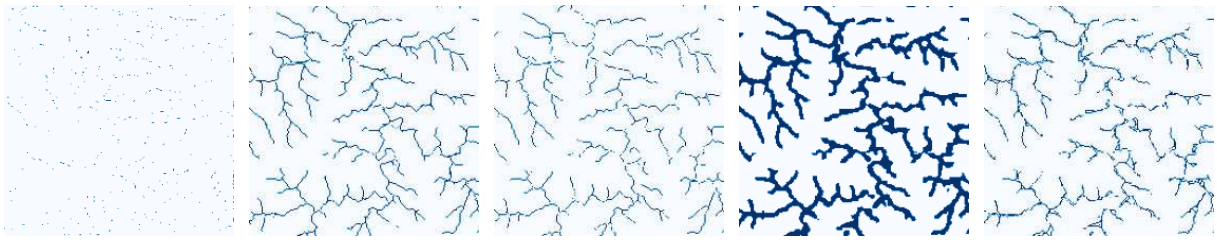
Figure 6. SRTM image sample, real label and dilated label.

decoder relies on an oversampling operation that scales the image dimensions from $16 \times 16 \rightarrow 128 \times 128$. This *upsamplig* operation impairs the ability to map very small objects within the image.

A solution found, was to dilate the drainage networks mask to perform the DeepLab V3 training. This expansion increases the drainage networks region in the training samples facilitating the networks mapping. A fully convolutional network (FCN) was trained, from the three images selected for training, to perform the erosion process of the drainage networks. The FCN showed an mean IoU of 95.8% in the erosion process.

Analyzing Table 1, in terms of accuracy and meanIOU, it can be seen that, among the deep learning models, DeepLab has the worst ability to identify drainage networks. A good identification can be observed on the dilated drainage networks comparing the predicted mask shown in figure 7(d) and the ground truth mask in figure 6(c). When performing the erosion process on the dilated predicted mask the final quality of the segmentation decreases, as can be seen in figure 7(e).

Finally, the Cycle GAN architecture presents an acceptable performance for this task, having a 92.43% in the IoU metric evaluation. This fact also can be observed by comparing the prediction made by Cycle GAN (figure 7(c)) and the reference used (figure 6(b)). Moreover, it was the architecture with the highest accuracy rates, despite having the lowest recall.



(a) Random Forest (b) U-Net (c) Cycle GAN (d) Deep Lab (e) FNC

Figure 7. Prediction of drainage networks performed

5 Conclusions

The proposed architectures have demonstrated the capability of using deep learning models for determining storm drainage networks through remote sensing images. Compared to the baseline used, the proposed models had a substantial gain over a more traditional classification technique like Random Forest that had its performance in terms of mean IoU equal to 54.5%. With an mean IoU higher than 90%, the techniques based on Deep Learning have shown to be powerful tools to perform the geoprocessing of large-scale images almost in real-time with good accuracy, a fact that can make the work of specialists more dynamic and efficient, thus bringing significant gains.

Acknowledgements. This study was financed in part by the Coordenação de Aperfeiçoamento de Pessoal de Nível Superior - Brasil (CAPES) - Finance Code 001. In addition, special thanks to the Laboratório Nacional de Computação Científica (LNCC) for the computational resources provided for this work.

References

[1] R. Jana, T. Reshmidevi, P. Arun, and T. Eldho. An enhanced technique in construction of the discrete drainage network from low-resolution spatial database. *Computers & geosciences*, vol. 33, n. 6, pp. 717–727, 2007.

[2] D. F. Maune. *Digital elevation model technologies and applications: the DEM users manual*. Asprs Publications, 2007.

[3] J. F. O'Callaghan and D. M. Mark. The extraction of drainage networks from digital elevation data. *Computer vision, graphics, and image processing*, vol. 28, n. 3, pp. 323–344, 1984.

[4] T. G. Freeman. Calculating catchment area with divergent flow based on a regular grid. *Computers & Geosciences*, vol. 17, n. 3, pp. 413–422, 1991.

[5] P. Quinn, K. Beven, P. Chevallier, and O. Planchon. The prediction of hillslope flow paths for distributed hydrological modelling using digital terrain models. *Hydrological processes*, vol. 5, n. 1, pp. 59–79, 1991.

[6] P. Holmgren. Multiple flow direction algorithms for runoff modelling in grid based elevation models: an empirical evaluation. *Hydrological processes*, vol. 8, n. 4, pp. 327–334, 1994.

[7] C. Qin, A.-X. Zhu, T. Pei, B. Li, C. Zhou, and L. Yang. An adaptive approach to selecting a flow-partition exponent for a multiple-flow-direction algorithm. *International Journal of Geographical Information Science*, vol. 21, n. 4, pp. 443–458, 2007.

[8] D. C. J. Fernández, de M. Morisson Valeriano, H. Zani, and de C. Oliveira Andrades Filho. Extração automática de redes de drenagem a partir de modelos digitais de elevação. *Revista Brasileira de Cartografia*, , n. 64/3, 2012.

[9] B. Zhao, J. Feng, X. Wu, and S. Yan. A survey on deep learning-based fine-grained object classification and semantic segmentation. *International Journal of Automation and Computing*, vol. 14, n. 2, pp. 119–135, 2017a.

[10] C. Donadio, M. Brescia, A. Riccardo, G. Angora, M. D. Veneri, and G. Riccio. A novel approach to the classification of terrestrial drainage networks based on deep learning and preliminary results on solar system bodies. *Scientific Reports*, vol. 11, n. 1, pp. 1–13, 2021.

[11] R. Kemker, C. Salvaggio, and C. Kanan. Algorithms for semantic segmentation of multispectral remote sensing imagery using deep learning. *ISPRS journal of photogrammetry and remote sensing*, vol. 145, pp. 60–77, 2018.

[12] F. F. Martins and M. C. Zaglia. Application of convolutional neural network to pixel-wise classification in deforestation detection using prodes data. In *GEOINFO*, pp. 57–65, 2019.

[13] N. Souly, C. Spampinato, and M. Shah. Semi supervised semantic segmentation using generative adversarial network. In *Proceedings of the IEEE International Conference on Computer Vision*, pp. 5688–5696, 2017.

[14] J. Long, E. Shelhamer, and T. Darrell. Fully convolutional networks for semantic segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 3431–3440, 2015.

[15] L.-C. Chen, G. Papandreou, I. Kokkinos, K. Murphy, and A. L. Yuille. Semantic image segmentation with deep convolutional nets and fully connected crfs. *arXiv preprint arXiv:1412.7062*, 2014.

[16] V. Badrinarayanan, A. Kendall, and R. Cipolla. Segnet: A deep convolutional encoder-decoder architecture for image segmentation. *IEEE transactions on pattern analysis and machine intelligence*, vol. 39, n. 12, pp. 2481–2495, 2017.

[17] O. Ronneberger, P. Fischer, and T. Brox. U-net: Convolutional networks for biomedical image segmentation. In *International Conference on Medical image computing and computer-assisted intervention*, pp. 234–241. Springer, 2015.

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

[19] H. Zhao, J. Shi, X. Qi, X. Wang, and J. Jia. Pyramid scene parsing network. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 2881–2890, 2017b.

[20] L.-C. Chen, G. Papandreou, I. Kokkinos, K. Murphy, and A. L. Yuille. Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs, 2016.

[21] I. J. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio. Generative adversarial networks, 2014.

[22] P. Isola, J.-Y. Zhu, T. Zhou, and A. A. Efros. Image-to-image translation with conditional adversarial networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 1125–1134, 2017.

[23] J.-Y. Zhu, T. Park, P. Isola, and A. A. Efros. Unpaired image-to-image translation using cycle-consistent adversarial networks. In *Proceedings of the IEEE international conference on computer vision*, pp. 2223–2232, 2017.

[24] M. A. Rahman and Y. Wang. Optimizing intersection-over-union in deep neural networks for image segmentation. In *International symposium on visual computing*, pp. 234–244. Springer, 2016.

[25] F. Chollet. Xception: Deep learning with depthwise separable convolutions. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 1251–1258, 2017.