



Article

Facing Erosion Identification in Railway Lines Using Pixel-Wise Deep-Based Approaches

Keiller Nogueira ^{1,2,*} , Gabriel L. S. Machado ¹ , Pedro H. T. Gama ¹ , Caio C. V. da Silva ¹ , Remis Balaniuk ^{3,4}, and Jefersson A. dos Santos ¹ 

¹ Department of Computer Science, Universidade Federal de Minas Gerais, Belo Horizonte-MG 31270-901, Brazil; gabriel.lucas@dcc.ufmg.br (G.L.S.M.); phtg@dcc.ufmg.br (P.H.T.G.); caiosilva@dcc.ufmg.br (C.C.V.d.S.); jefersson@dcc.ufmg.br (J.A.d.S.)

² Computing Science and Mathematics, University of Stirling, Stirling, Scotland FK9 4LA, UK

³ Universidade Católica de Brasília, Taguatinga, Brasília-DF 71966-700, Brazil; remisb@tcu.gov.br

⁴ Tribunal de Contas da União (TCU), Setor de Administração Federal Sul (SAFS), Brasília-DF 70042-900, Brazil

* Correspondence: keiller.nogueira@dcc.ufmg.br or kno@cs.stir.uk; Tel.: +44-1786-46-7435

Received: 4 December 2019; Accepted: 6 February 2020; Published: 23 February 2020



Abstract: Soil erosion is considered one of the most expensive natural hazards with a high impact on several infrastructure assets. Among them, railway lines are one of the most likely constructions for the appearance of erosion and, consequently, one of the most troublesome due to the maintenance costs, risks of derailments, and so on. Therefore, it is fundamental to identify and monitor erosion in railway lines to prevent major consequences. Currently, erosion identification is manually performed by humans using huge image sets, a time-consuming and slow task. Hence, automatic machine learning methods appear as an appealing alternative. A crucial step for automatic erosion identification is to create a good feature representation. Towards such objective, deep learning can learn data-driven features and classifiers. In this paper, we propose a novel deep learning-based framework capable of performing erosion identification in railway lines. Six techniques were evaluated and the best one, Dynamic Dilated ConvNet, was integrated into this framework that was then encapsulated into a new ArcGIS plugin to facilitate its use by non-programmer users. To analyze such techniques, we also propose a new dataset, composed of almost 2000 high-resolution images.

Keywords: deep learning; remote sensing; erosion identification; high-resolution images

1. Introduction

Soil erosion, defined as the detachment, transportation, and deposition of soil by water or wind, is the most important land degradation problem worldwide [1]. It is so relevant that, in the past century, an increasing amount of research has been published focusing on erosion identification due to its ubiquity and severity [2]. Moreover, erosion may have a severe impact on infrastructure assets, which may culminate in an extremely alarming situation that, in turn, can result in economic losses and, in the worst case, human casualties [3]. Among large infrastructures, railway lines are one of the most likely constructions for the appearance of erosion due to the enormous amount of infrastructure slopes (throughout their entire huge extension), such as embankments and soil cuttings that are particularly vulnerable after prolonged periods of wet weather or more intensive short duration rainfall events. This is especially worrying because the structural safety of the railway is directly connected to the slope stability of embankments, which, in turn, is determined by the degree of damage suffered by them due to erosion. Hence, unaware or unsupervised erosion may get worse over time resulting in acute problems, such as cuts, rills, and gullies, which may culminate in even more impactful issues, including risks of derailments, possible interruptions of normal train operations, and environmental degradation.

Therefore, it is fundamental to early identify and monitor erosion in railway lines in order to prevent major consequences. Typically, such identification and monitoring are performed *in loco*, but the accessibility to high-resolution aerial imagery has allowed for analyzing larger areas quickly and with less effort. In this case, erosion identification is conducted by specialists that must visually recognize them into huge aerial photography datasets by examining every image. It is a faster process than monitoring physically, but still a time-consuming and slow task. Thus, with the purpose of trying to speed up the process, automatic methods to perform erosion identification using remote sensing images appear as an economic and appealing alternative for the society.

Towards such an objective, during the last few decades, researchers have proposed and experimented distinct machine learning-based approaches to automatically perform erosion identification in aerial images. In all those methods, the feature extraction, responsible for quantitatively describing the images based only on its internal pixels, is a fundamental step, given that spatial feature representation in a suitable way is the key for generating good pattern classifiers. However, extracting meaningful features of erosion areas is a complicated task given the: (i) high variation in size, since an erosion can vary from few centimeters to meters in both length and depth, (ii) distinct shapes and textures, as an erosion can occur in many different patterns and places, (iii) high temporal variability, since an erosion may vary notably with the action of time, and (iv) high difference in illumination and shadows, common aspects in remote sensing images. Hence, as important as automating the process is creating a good feature representation for the remote sensing images in order to allow the machine learning approaches to capture all feasible information and, then, accurately perform erosion identification.

Focused on creating good feature representation, a prominent technique, called deep learning [4], can learn the adaptable and specific features and the classifiers in a single training stage (end-to-end). Deep Learning is a sub-field of machine learning that primarily focuses on multi-layered neural network models. These models have, inherently, a feature learning step that allows a better data-driven encoding of feature. This process of feature learning is performed automatically by the models, thus eliminating the necessity of manually setup feature extraction algorithms. Convolutional Network (ConvNet) [4] is the most common deep learning method for learning visual features in computer vision applications, including remote sensing ones [5]. The success of this network for image related applications is mainly due to the fact that it exploits the natural stationary property of the images, i.e., the statistics of one part of the image are assumed to be the same as those of any other part [6]. ConvNets are able to learn and combine features from different levels of abstraction: (i) low-level information (such as corners and edges) is captured in the first layers; (ii) intermediate layers are capable of learning mid-level patterns (as object parts); and (iii) high-level information (such as whole objects) are learned by the last layers.

Given the benefits of deep learning, the main objective of this work is to develop an effective tool (based on supervised learning) for erosion identification. Towards such goal, we introduce a new framework, based on ConvNets, to perform supervised erosion identification in railway lines using high-resolution aerial image sets. Precisely, we tackle the erosion identification as a (semantic) segmentation (also known as pixel classification) task, in which each pixel of the input is labeled into one of the classes. Although more difficult, such task allows a better definition of the erosion boundaries which can lead up to other useful information, such as the exact land area covered by the erosion, a possible estimate of the price to fix the problem, and others. All this knowledge can be used by agents to propose better plans of actions, define priorities, and so on.

In order to define the best deep learning segmentation technique to be integrated into the proposed framework, six successful supervised approaches were evaluated in this work, including: (i) Pixelwise [7], a segmentation algorithm that classifies each pixel independently using context windows, (ii) Fully Convolutional Network (FCN) [8], one of the first fully convolution architectures proposed for pixel labeling, (iii) Deconvolution network [9–11], an encoder–decoder fully convolutional architecture proposed for semantic segmentation, (iv) DeepLabV3+ [12], a multi-scale semantic

segmentation approach that combines atrous spatial pyramid pooling with an encoder–decoder architecture based on depthwise separable convolutions, (v) Dynamic Dilated ConvNet (DDCN) [5], a pixel labeling approach based on dilated convolutions that exploits multi-scale information by allowing the training with dynamic patch sizes, and (vi) Mask R-CNN [13], an instance segmentation technique which proposes regions and then annotates each pixel of such proposals by using a fully convolutional network.

Although the proposed framework can be used as standalone package, we encapsulate it as a plugin for the software ArcGIS (<https://www.arcgis.com/>). The main advantage of using the proposed framework as an ArcGIS plugin is the possibility of combining the state-of-the-art technique (embedded into the plugin) with the already implemented visualization, manipulation, and processing tools of ArcGIS. Note that the source-codes for the framework and for the ArcGIS tool are publicly available (<https://github.com/Gabriellm2003/Deep-Learning-ArcGIS-Plugin>).

In summary, we claim the following contributions for this work:

- a novel Railway Erosion dataset composed of 1960 448×448 high-resolution satellite images,
- an evaluation and analysis of six deep learning-based methods to perform erosion identification in an aerial high-resolution remote sensing dataset,
- a new tool for erosion identification using satellite imagery,
- a new plugin for the ArcGIS software based on the proposed tool.

It is important to emphasize that, as far as the authors are aware, this is the first work to use deep learning-based techniques to perform erosion identification in remote sensing images.

The remainder of this paper is structured as follows. Related works are presented in Section 2. Section 3 presents, in detail, all deep learning-based techniques evaluated in this work. In Section 4, we present the framework and ArcGIS plugin. The experimental protocol for the evaluation of the methods is presented in Section 5. Section 6 presents the produced results while Section 7 discusses such outcomes. Finally, Section 8 concludes the work, pointing out the findings and future work.

2. Related Work

The increasing availability of high-resolution aerial imagery (provided by new sensor technologies) allows the deployment of a growing number of remote sensing datasets which, in turn, opens new opportunities for Earth Observation applications, such as erosion identification. However, as introduced, performing such task using manual efforts is costly and automatic methods appear as an attractive alternative. In fact, over the years, several techniques [3,14–23] have been proposed to perform erosion identification using remote sensing datasets.

Benzer [14] exploited distinct factors—including rainfall erosivity, soil erodibility, and vegetative cover—to perform erosion identification. Particularly, those factors, which consist of a set of logically related geographic features and attributes, were combined using the RUSLE model [24] in order to finally perform the erosion identification. In [15], the authors perform soil erosion identification and monitoring using multi-temporal and multi-resolution Digital Terrain Models (DTMs) produced from images captured by Unmanned Aerial Vehicles (UAVs). They calculated the difference between the DTMs in two different points in time to understand and monitor the gully volume change. Eltner et al. [16] also used high precision DTM, extracted from UAV as well as from terrestrial laser scanning (TLS), to perform multi-temporal change detection of erosion areas.

More recently, Liu et al. [3] proposed a system to perform gully erosion identification in catchments using high-resolution Digital Elevation Models (DEMs) and ortho-mosaics produced from images captured by UAV. Their approach combines an object segmentation algorithm, called Multi-resolution Image Segmentation Algorithm (MRIS) [25], with random forest to perform erosion identification. In [17], the authors classified different soil erosion types using terrain and spectral attributes obtained from hyperspectral images. Different machine learning algorithms—such as Support Vector Machine (SVM), random forest, and shallow Artificial Neural Network (ANN)—are trained to predict the type

of soil erosion. Arif et al. [18] used a simple Multi-Layer Perceptron (MLP) network to perform erosion classification. Such network receives as input five factors of erosion control—erosivity, erodibility, length and slope, land cover management, land conservation practice, and the spectral signature of a SPOT 5 multispectral image with four bands/channels and outputs the erosion classification. Krenz et al. [19] exploited vegetation images and topographic information, also captured by a UAV, to identify areas of soil degradation. In [20], authors proposed to employ different indexes—such as fractional vegetation coverage, nitrogen reflectance index, yellow leaf index, and bare soil index—to perform erosion detection. Specifically, such indexes, derived from remote sensing imagery, were integrated to create a final model through Principal Component Analysis (PCA). In [21], the authors exploited multiple indexes, computed over MODIS data, to perform erosion assessment. Such indexes were used to train a random forest model that is responsible to perform the final erosion mapping. Gianinetto et al. [22] proposed the dynamic version of the RUSLE model [24]. In fact, the original formulation was altered to incorporate variations in rainfall erosivity and land-cover allowing the estimation of both spatial and temporal land-cover changes. Peponi et al. [23] combined geographic information systems and shallow ANNs to design a model that forecasts the erosion changes using satellite images.

Our work is the first one in erosion recognition that focuses on the segmentation of erosion areas in railway lines using high-resolution aerial image sets. Furthermore, although some of the previous works [17,18,23] employ artificial networks to tackle the problem, none of them use deep learning-based approaches to perform erosion segmentation. In fact, as far as we know, this is the first work to use deep learning for erosion segmentation in high-resolution remote sensing images.

3. Background

This section aims to explain all deep learning methods that were evaluated in this work for erosion identification. Such semantic segmentation approaches were selected based on their popularity and performance for different applications and images, including computer vision [8–13], remote sensing [5,7,26–30], medical [31–35], and so on. However, although their success in different domains, as aforementioned, those methods were never evaluated in the erosion identification task.

Section 3.1 describes the pixelwise [7] strategy and its architecture. Fully Convolutional Network (FCN) [8] is presented in Section 3.2, while Section 3.3 introduces the concept of Deconvolution Network and its architecture [9,10]. In Section 3.4, the famous fully convolutional DeepLab [36] architecture is detailed. Section 3.6 presents the Mask R-CNN [13] and its idea of performing object detection and segmentation at the same time. Finally, Section 3.5 explains the Dynamic Dilated ConvNet (DDCN) concept and architecture.

3.1. Pixelwise Network

The first deep learning-based strategy evaluated in this work is the pixelwise one [7]. In such technique, each pixel of the input image is classified independently. Precisely, each pixel is represented by a context window, i.e., overlapping fixed-size patches, in which each one is centered on a specific pixel helping to understand the spatial patterns around that pixel. Observe that these context windows are really necessary because the pixel itself has not enough information to be used in its classification. Such patches are, in fact, used to train and evaluate the network. In both processes, the ConvNet outputs a class for each input context window, which is associated with the central pixel of the window. The main advantage of this method is the possibility of classifying each and every pixel based on its context, which creates detailed high-resolution outputs [7]. However, the computational resources required for these techniques are huge since every pixel of the image becomes a context window for the network.

An overview of the network architecture, based on the pixelwise strategy, evaluated in this work is presented in Figure 1. Such network, proposed by [7], receives as input 25×25 context windows and has three convolutional and max-pooling layers, and two fully-connected ones. Furthermore, after each fully-connected layer, there is a dropout [4] with a 50% chance of randomly dropping a neuron. It is important to emphasize that, because of its advantages [7], Rectified Linear Unit (ReLU) [4] was the processing units used in all layers of this ConvNet.

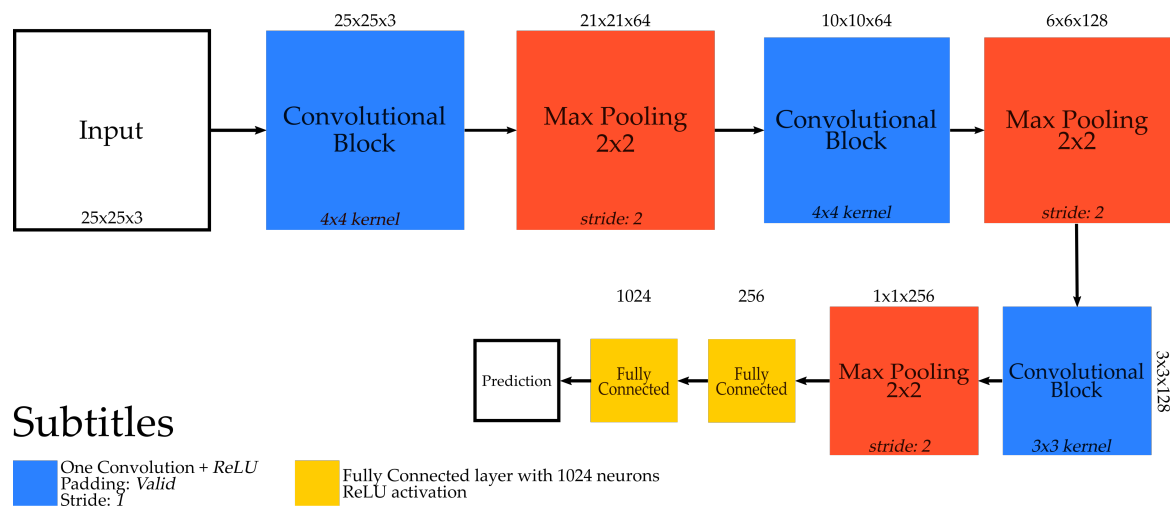


Figure 1. Pixelwise Network architecture [7].

3.2. Fully Convolutional Network (FCN)

Towards a precise semantic segmentation using less computational resources, there is the Fully Convolutional Network (FCN) [8]. Such network takes as input the original image and returns, as output, another image (i.e., the dense prediction), with the same resolution of the input, but with each pixel associated with a class. This is possible due to the use of deconvolution layers [37] that learn how to upsample the feature maps and produce the final dense prediction.

Figure 2 presents the FCN architecture evaluated in this work. This network is very similar to standard ConvNets (such as the one proposed in the previous section) but has subtle modifications. Particularly, this FCN does not have fully-connected layers being essentially composed of convolution and max-pooling ones. This allows the network to preserve the spatiality of the feature maps (since fully-connected layers lose such property), which is required in order to perform finer dense classification. In addition, as aforementioned, this network uses deconvolution layers [37] to upsample the prediction into pixel-dense outputs. This is essential because the convolution and max-pooling layers reduce the resolution of the input, whereas the deconvolution ones restore the spatial size outputting a prediction with the same height and width of the input image.

Aside from this, note the use of elementwise addition in this architecture. This is explored because creating the final prediction map using only the classification performed by the last convolution may result in deceptive prediction maps [8]. Thus, to overcome this problem, skip connections were incorporated into the architecture (usually coming from pooling layers). These connections allow the network to combine (via elementwise addition) prediction maps (created from distinct feature maps) that combine information from fine (final) and coarse (initial) layers, allowing the model to make local predictions that respect the global image structure.

The obvious advantage of this method is a reduced computational complexity, given that it classifies patches instead of pixels. Furthermore, the prediction maps can have distinct resolution for training and testing, i.e., the size of the training images can be different from the testing ones which gives more independence for the ConvNet. Although using one deconvolutional layer to upsample the output of a ConvNet provides dense outputs, the result may be imprecise because the upsampling process performed is too simple. Thus, usually more than one upsample operation is used, as in the case of the evaluated architecture. Even though this model uses two deconvolutions, the direct upsample to the output forces the last layer to learn the transformation from a smaller space to a larger, at the same time that is directed influenced by the prediction. This can create undesirable results once the layer has to focus in two different tasks.

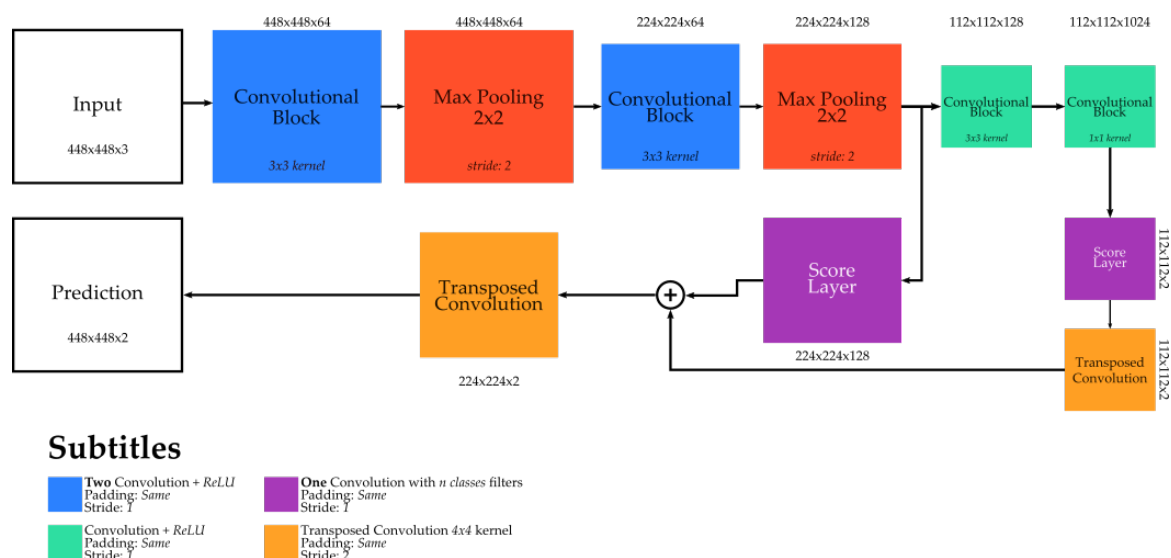


Figure 2. Fully Convolutional Network architecture [8].

3.3. Deconvolution Network

Deconvolutional networks [9–11] try to overcome the aforementioned issue by using multiple deconvolutional layers [37] to produce the final dense prediction (from a coarse feature map) with the same resolution of the input image. Normally, such networks are based on encoder–decoder architecture. The encoder works as a fully ConvNet (without the final classification layer) by running the following pipeline: it receives input images; learns the visual features by using standard convolution and max-pooling layers; and outputs a coarse feature map. The decoder receives a coarse map generated by the encoder as input. It learns to upsample these features by using deconvolution layers to increase the spatial resolution of the coarse map gradually. The decoder output is a prediction map with the same dimensions ($h \times w$) of the original input image. The encoder–decoder functions as one larger model. That is, they are trained together in an end-to-end manner by using classical feedforward and backpropagation algorithms, and are also in used/tested in a connected fashion.

Figure 3 presents the deconvolutional network architecture exploited in this work. The encoder part of this network is composed of three convolution layers (each one followed by a max-pooling). It receives the input image and outputs a 56×56 coarse feature map. The decoder part has three deconvolution layers (each one followed by a convolution layer), very similar to [11,12]. This part receives the coarse feature map (produced by the encoder) and outputs the final prediction image.

The biggest advantage of this technique relies on the use of more than one deconvolution layer. It gradually improves the spatial information of the outcome resulting in smoother predictions than the fully convolutional networks [8]. On the other hand, the disadvantage of such approach is the training load. Since the network is significantly larger, the optimization may be slow and difficult due to the huge number of trainable parameters.

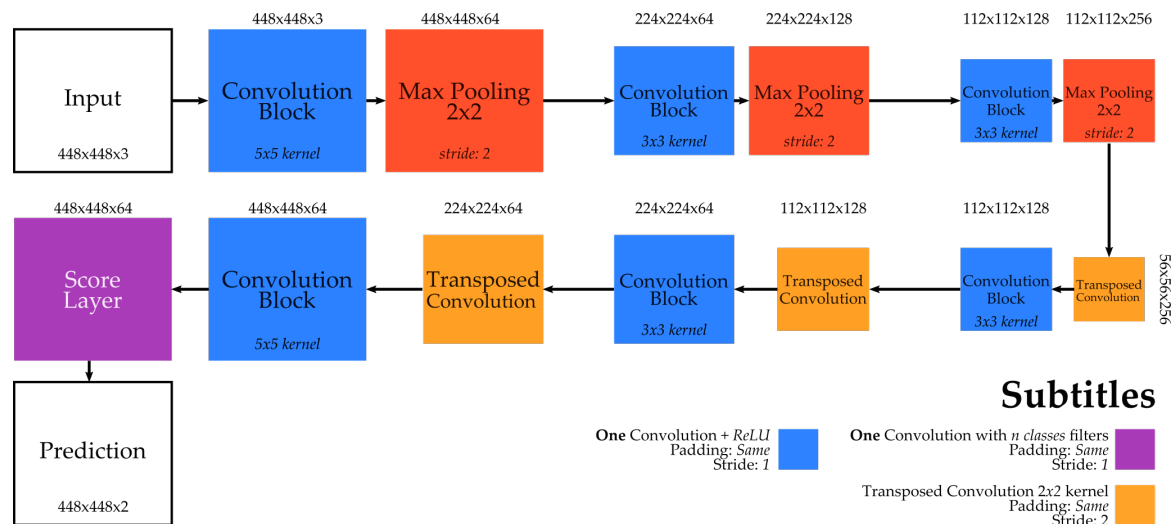


Figure 3. Encoder–Decoder Deconvolutional architecture [9–11].

3.4. DeepLabV3+

Recently, researchers [5,12,38–41] noticed that smoother predictions could be generated if the input image resolution was not reduced substantially, i.e., if the input information was preserved over the layers. However, this comes with a problem: by preserving data resolution, the networks would not be able to exploit certain benefits that they are capable of when downsampling the data, such as the increase of the receptive field (and, consequently, of the observable area) [4]. To overcome such a problem, dilated convolutions [42] were proposed. Such layers are capable of increasing the receptive field whereas preserving the resolution, i.e., no downsampling in the data are performed.

Over the years, several methods have been proposed to exploit the advantages of dilated convolutions [5,12,38–41]. Among such approaches, one of the first (and most famous) techniques proposed for semantic segmentation is the DeepLab [12,38,40,41]. This method, based on fully convolution networks, has several versions (V1 [38], V2 [40], V3 [41], and V3+ [12]) that exploits the same concept: the input resolution is downsampled in the first layers from which it is kept constant by the final dilated convolutions. Essentially, the differences between the versions are the use of more dilated convolutions to make the method more robust.

In this work, we evaluated the DeepLabV3+ version [12], whose architecture is presented in Figure 4. Technically, such network can be seen as an encoder–decoder architecture.

In the encoder part, it first uses a (usually pre-trained) network to extract low-level features. Different networks can be used as a backbone to extract the features, such as Xception [43] and MobileNetV2 [36]. In this work, we use a **pre-trained** (on VOC Pascal dataset [44]) **MobileNetV2** network [36], in which 1280 56×56 feature maps are extracted from the layer just before the classification one. Such features are further processed, in the encoder part, by an Atrous Spatial Pyramid Pooling (ASPP) module. This component is composed of a set of atrous convolution layers [42] that process the same input features but with different dilation rates, allowing it to capture and aggregate multi-scale information.

The decoder part concatenates the low-level features (extracted by the Feature Extractor module) with (a bilinear upsampled version of) the multi-scale features extracted by the ASPP module. Then, it further processes the concatenated features using standard convolution layers. Finally, a simple bilinear upsampling is performed to retrieve feature maps with the same resolution of the input data that are then, processed using a 1×1 convolution layer, producing the final prediction map.

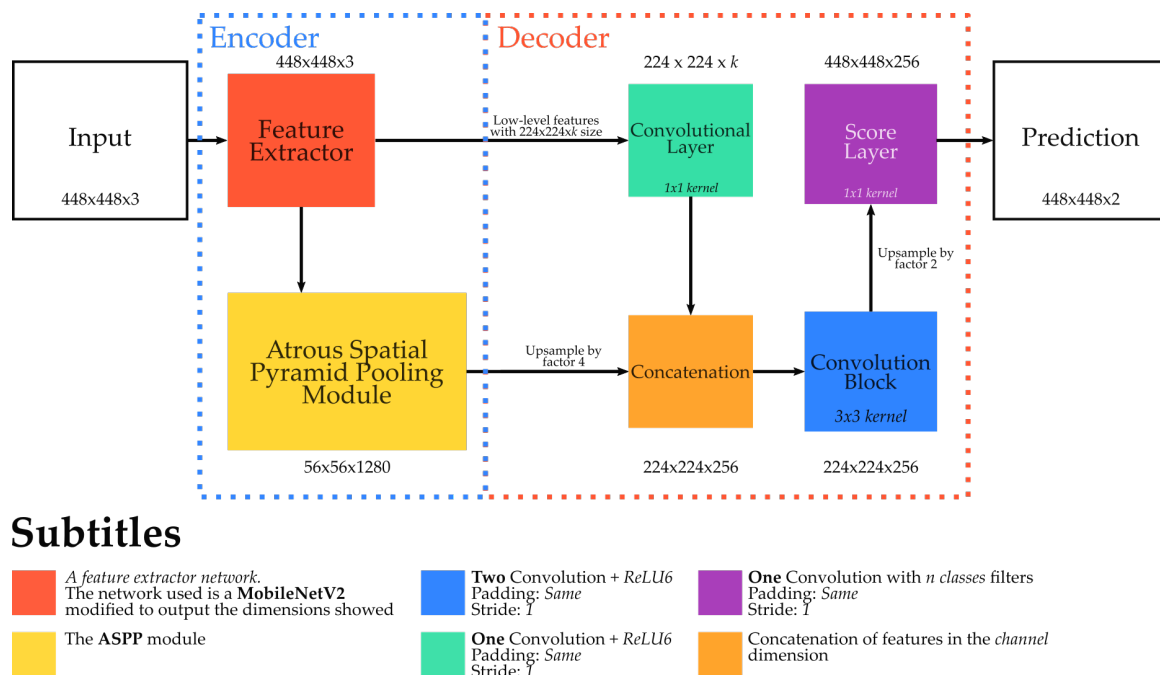


Figure 4. DeepLabV3+ architecture [12].

An advantage of the DeepLabV3+ network [12] is its capacity of generating a high-quality segmentation mask, which is achieved by not reducing the initial resolution substantially. The disadvantage would be that the input data are still downsampled, in the first layers of the backbone network, resulting in a likely loss of valuable information.

3.5. Dynamic Dilated ConvNet

To try to overcome previous drawback, a recent method, called Dynamic Dilated ConvNet (DDCN) [5], takes the concept of preserving the input image resolution to the extreme. Specifically, this semantic segmentation technique proposes a new multi-scale training strategy that employs dynamic input sizes to converge a fully dilated convolution network that never reduces the input image.

The method receives as input the images and distributions over the possible sizes that the network input image (i.e, patch) may have. In this work, we used a uniform distribution that allows the method to select an input size from three possibilities: 50×50 , 75×75 , and 100×100 . In each iteration of the training, an input size is randomly selected from the given distribution and a new batch composed of distinct images (when compared to other batches) of the pre-selected input resolution is created and used to converge the model. It is important to note that the multi-scale information is learned during the model training by composite batches of inputs with multiple sizes. In the prediction phase, the algorithm selects, based on scores accumulated during the training for each evaluated input size, the best resolution for a given problem. The technique processes the testing images using batches composed of images with the best-evaluated resolution.

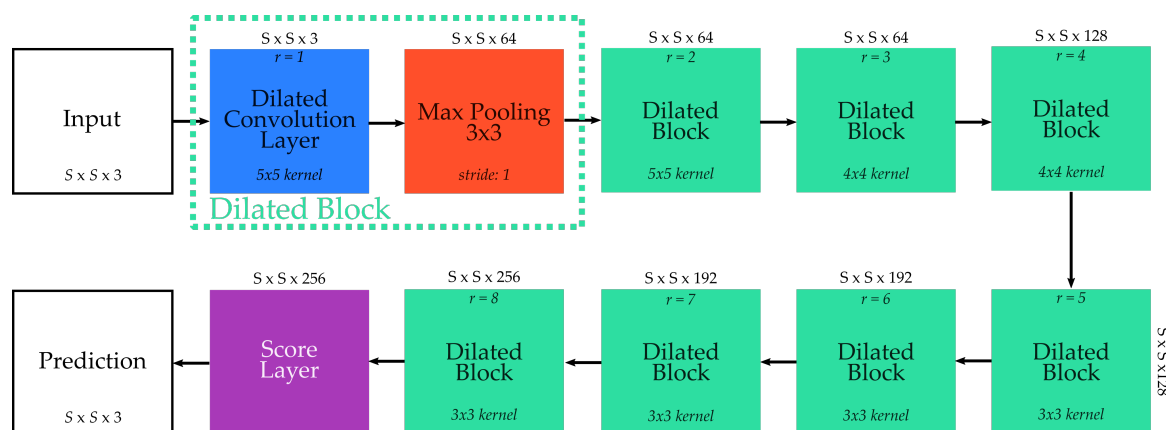
Several different architectures could be used with this training strategy, such as fully convolution [8,12,38,40,41] and deconvolution networks [9–11]. Although these networks may be able to process inputs of varying sizes, they may have problems related to a possible high variation of the input size. Precisely, such networks require the input to be large enough so that they can downsample it to generate a coarse map, which is then upsampled to the original input size. The network may not be able to generate this coarse map if the input is too small.

To overcome this problem, the authors in [5] proposed a new network composed uniquely of dilated convolutions [42] that never reduces the input image. This fully dilated network fits perfectly into the proposed multi-scale training strategy as it is capable of processing inputs of any size (without

constraints since no downsampling would be performed at all) while still outputting results with the same resolution of the input data. Note that, unlike DeepLabV3+ [12], this network never reduces the input image, preserving its resolution from end-to-end, allowing the model to extract more useful information and, consequently, it can produce smoother predictions.

The fully dilated network architecture evaluated in this work is presented in Figure 5. It has eight dilated blocks, each one composed of a dilated convolution and a max-pooling layer. Besides that, it is worth mentioning that the pooling layers do not downsample the input given with a specific configuration of stride and padding (i.e., stride 1 and padding same).

Specifically, the dilated convolutions of first two blocks have 5×5 filters with dilation rates 1 and 2, respectively. The convolution layers of the following two dilated blocks have 4×4 kernel filters but rates 3 and 4, correspondingly. The last four dilated blocks have convolution layers with smaller filters (3×3) but higher dilation rates (5, 6, 7 and 8, respectively). Finally, after all those layers, there is a final 1×1 convolution layer responsible for generating the final dense predictions.



Subtitles

Dilated Convolution + ReLU where r is the dilation rate
Padding: Same
Stride: 1

One Convolution with n classes filters
Padding: Same
Stride: 1

S is selected from the set {50, 100, 150}

Figure 5. Dynamic Dilated Network architecture [5].

Looking at pros and cons, an advantage of the DDCN [5] is the possibility of exploiting multi-scale information without increasing the complexity of the network while preserving the initial image resolution, which can help the network to learn even more representative patterns from the data. The main disadvantage of this approach is the training time. Since the input image is not downsampled, the model needs more time to converge as the processing time of each layer is proportional to the input data resolution [45].

3.6. Mask R-CNN

Even more recently, new approaches for performing object identification were conceived based on the concept of instance segmentation. Differently from semantic segmentation techniques (such as the previous ones presented in this Section), which output dense predictions inferring a class for every pixel of the input image, instance segmentation techniques try to first identify and locate the object instance (delineating its boundaries), and after that they segment the object of interest. Therefore, the final prediction of instance segmentation approaches may not be totally dense, i.e., only the interested objects will be identified and segmented in the final outcome, leaving all the uninterested parts of the input image without a label classification.

In this work, we evaluated the Mask R-CNN [13], an instance segmentation technique that detects and segment pixels of the object instances by using a Fully Convolutional Network. Technically, the Mask R-CNN [13] is strongly based on the Faster R-CNN [46], an object detection approach that receives input images, detects the object instances, and outputs the bounding boxes proposals for object detection. The Mask R-CNN technique [13] uses the Faster R-CNN [46] to first generate the object instance bounding boxes and then segments such proposals to produce the final prediction map for each object instance. Overall, the Mask R-CNN [13] efficiently detects object instances in an image and simultaneously generates a high-quality thematic map for each detection by adding just a small computational overhead to the Faster R-CNN [46].

An overview of the Mask R-CNN architecture [13] is presented in Figure 6. The first module of this architecture, called here CNN, is responsible to extract the features and create a representation for the input images. In this work, this module is actually composed of a **ResNet-101** [47], **pre-trained** on the ImageNet dataset [48]. Then, a module, called *Region Proposal Network* (RPN), is responsible to create bounding proposals, while the module *Region of Interest (RoI) align* performs the alignment between the feature maps (extracted by the CNN module) with the proposals. Then, the network architecture divides into two branches. The first one, composed of the *Fully Connected (FC) Layers*, is responsible to detect and classify the proposals while the second one, composed of *Convolution Block* and *Score Layer*, classifies the pixels of the proposals, segmenting the objects. Note that, except for the second branch of the last part, all other modules are part of the original Faster R-CNN architecture [46].

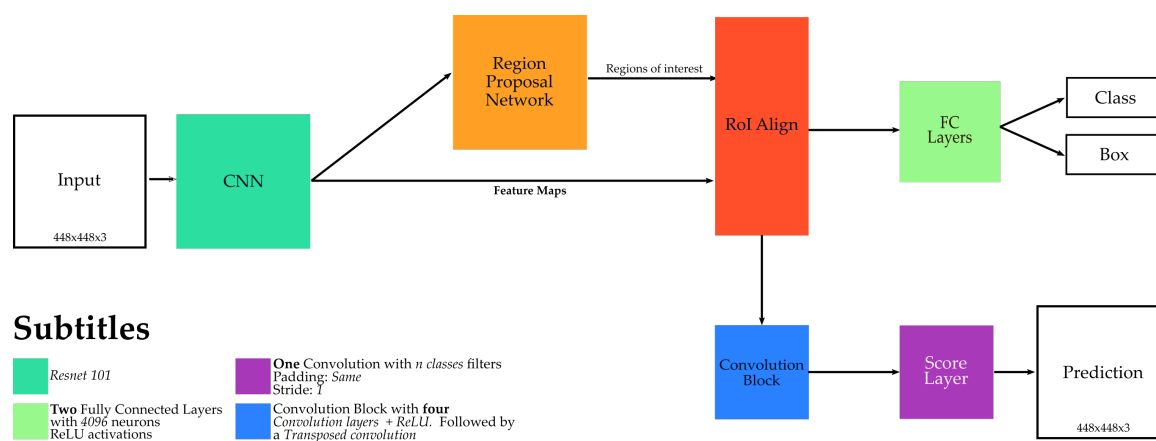


Figure 6. Mask R-CNN architecture [13].

An advantage of the Mask R-CNN [13] (when compared to other methods) is its capacity of generating high-quality segmentation mask for each object, since it classifies each instance separately. On the other hand, one disadvantage of this approach may be the training. Even using a pre-trained network in the CNN module, Mask R-CNN [13] has several other modules (each with lots of parameters) that needed to be trained. Therefore, the training of this technique may not be that easy.

4. Deep Learning for Erosion Identification

In this section, we detail the proposed framework and the ArcGIS Plugin. Specifically, an overview of the framework is given in Section 4.1, while the plugin is presented in Section 4.2.

4.1. Framework

The proposed framework, presented in Figure 7, receives as input the remote sensing images and a georeferenced railway shapefile. Since we are interested in identifying erosion areas along railway lines, the first step of the proposed framework is to extract overlapping 448×448 crops from the input images based on the railway location, i.e., to extract patches on which the railway is centered. By doing this, the framework becomes more robust to process large-scale remote sensing datasets and their huge

images, given that it keeps the focus on relevant areas of the original input images (the ones close to the railway) while discarding useless parts of such data (distant areas of the railroad).

The next step of the proposed framework is to process those crops using deep learning-based methods, responsible for the identification of the erosion areas. In fact, in this step, a pre-trained neural network outputs a prediction image for each input crop (with the same resolution). Note that it is possible to perform two operations over this neural network in the proposed framework: fine-tuning and inference. The only difference between these operations is that, in the latter one, all crops are classified (generating prediction images), whereas, in the former one, part of the extracted crops is first used to perform small adjustments and improve the network that is, then, explored to classify the remaining crops. Furthermore, although those deep learning processes can be considered very time-consuming [4], the first step of the framework (that discard useless—far from the railway—data) makes them relatively quicker. It is also important to observe that the framework was designed to be flexible and adaptive, as distinct deep learning-based techniques can be easily incorporated into the proposed tool.

The last step of the framework uses the georeference of the original input data to merge all prediction crops into a final georeferenced prediction image (with the same resolution of the original input data). Since these prediction crops overlap, the merge process is performed evaluating each pixel separately. In this case, if a predicted pixel does not overlap with any other crop, the final prediction image will preserve the original class predicted for that pixel. However, if a predicted pixel overlaps with other crops, then an average (for all overlapping pixels) is calculated (with respect to the classes) and the class with the highest confidence is propagated for the final prediction image. After this merge process, the prediction image is converted into a georeferenced shapefile that contains all the predictions represented by a georeferenced polygon and that can be easily exploited in any Geographic Information System, such as ArcGIS.

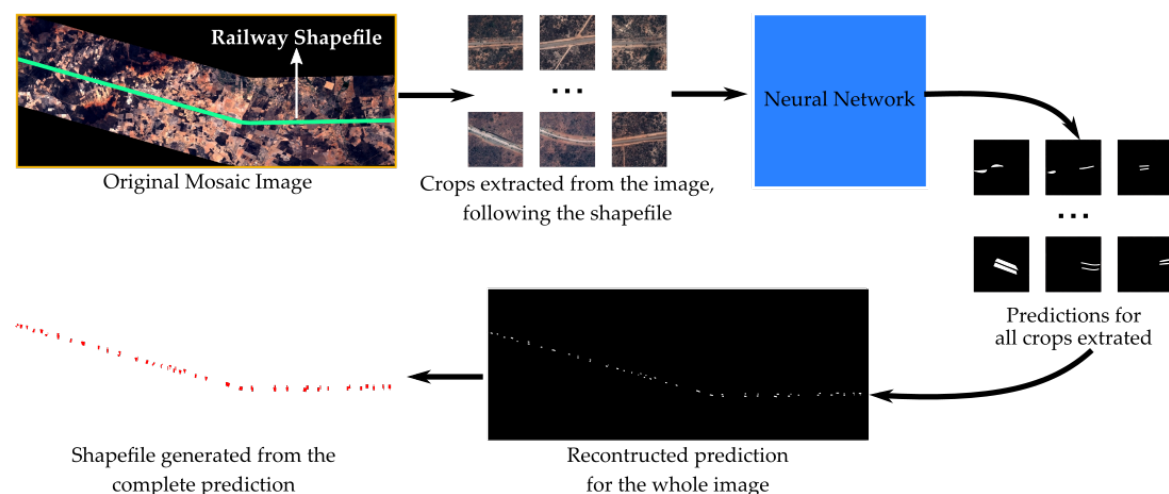
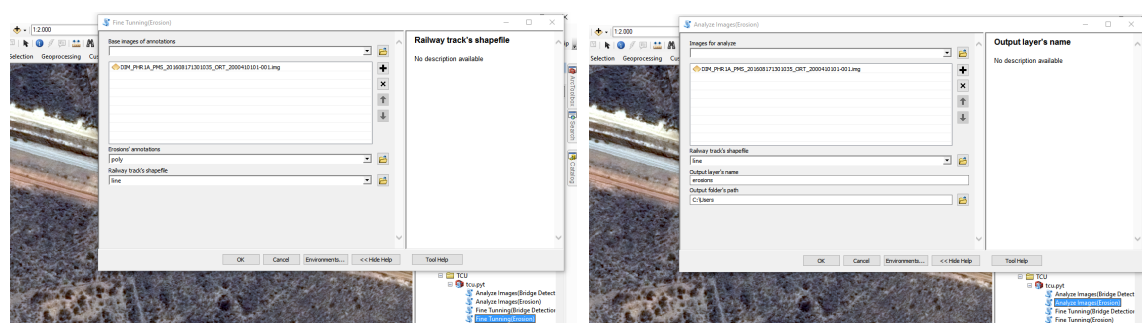


Figure 7. The proposed deep learning framework for erosion identification in large-scale remote sensing datasets.

4.2. ArcGIS Plugin

As introduced, the proposed framework was incorporated into an ArcGIS framework in order to facilitate its use by non-programmer end-users. Precisely, this plugin encapsulates the proposed deep learning-based framework allowing the user to exploit it for both training and inference. The plugin window of these two operations can be seen in Figure 8.

Observe that, in order to exploit the benefits of the plugin, the end-user just needs to set some fields (with file paths) and execute them (All information about the plugin, its fields, and how to set them can be found at <https://github.com/Gabriellm2003/Deep-Learning-ArcGIS-Plugin>). When the execution of the framework is finished, a shapefile, showing the identified erosion areas, is created and loaded directly into the ArcGIS software, requiring no efforts from the end-user. This simple configuration and execution show how easy is to exploit the benefits of the proposed plugin, given that no programming or advanced ArcGIS skills are required.



(a) Train Plugin Window

(b) Inference Plugin Window

Figure 8. Train and inference windows of the proposed ArcGIS plugin for erosion identification.

5. Experimental Configuration

In this section we present, in detail, the experimental configuration employed to train and compare the aforementioned techniques. Precisely, Section 5.1 presents the Railway Erosion Dataset, proposed and exploited in this paper. The experimental protocol is described in Section 5.2 while software and hardware information are presented in Section 5.3.

5.1. Railway Erosion Dataset

The proposed dataset, publicly available (https://drive.google.com/file/d/1LE9tFt3VMka9hQtGnd7QkAQhZsU_COEa/view?usp=sharing), is composed of image crops extracted from a georeferenced mosaic of 53 high-resolution satellite imagery covering the 1753 km of the *Transnordestina Railway* in Brazil. These aerial images, captured by the *Plêiades* satellite, have an average size of $39,700 \times 30,939$ pixels (requiring around 2TB for storage) and were originally obtained by the Brazilian Federal Court of Accounts (Tribunal de Contas da União—TCU), in partnership with the German Agency for International Cooperation (GIZ). Each image is composed of four channels (RGB and near-infrared bands, in this order) and has 0.5 m of spatial resolution.

All of these data were visually and carefully analyzed and the erosion areas, in the vicinity of the railroad, were manually annotated by specialists (geographers and geologists from the University of Brasilia—UnB). For each occurrence of erosion found, a georeferenced polygon was designed in order to identify its borders, generating the segmentation masks (i.e., the class labels of the images). An example of an image with a highlight of an erosion area and its respective mask is presented in Figure 9.

The Railway Erosion dataset is actually composed of 1960 448×448 image crops of the annotated erosion areas of this large-scale mosaic over the *Transnordestina Railway*. All images of this dataset were normalized by subtracting the mean and dividing by the standard deviation. Some examples of these images are presented in Figure 10.



Figure 9. A sample image of *Railway Erosion Dataset* and a zoomed region with ground-truth. Legend—White: Erosion areas. Black: Non-erosion areas/Background.

5.2. Experimental Protocol

To train and compare the methods introduced in Section 4, we carried out a 5-fold cross-validation over the Railway Erosion dataset. Following this strategy, the dataset was randomly split into five mutually exclusive subsets/folds. Then, five runs were executed, where, at each run, three folds were used as training, one as validation, and the remaining as test. Note that each run requires the training of a new network. The reported results are the average metrics of the five runs followed by its corresponding standard deviation.

In this work, results are reported in terms of three pixel-level metrics: (i) Normalized (or Average) Accuracy [49] that reports the average of a per-class accuracy, (ii) Intersection over Union—IoU (also known as Jaccard Index) [49] that essentially quantifies the overlap between the predicted outcome and the ground-truth, and (iii) Kappa [50] that also measures the agreement between the classified outcome and the reference data.

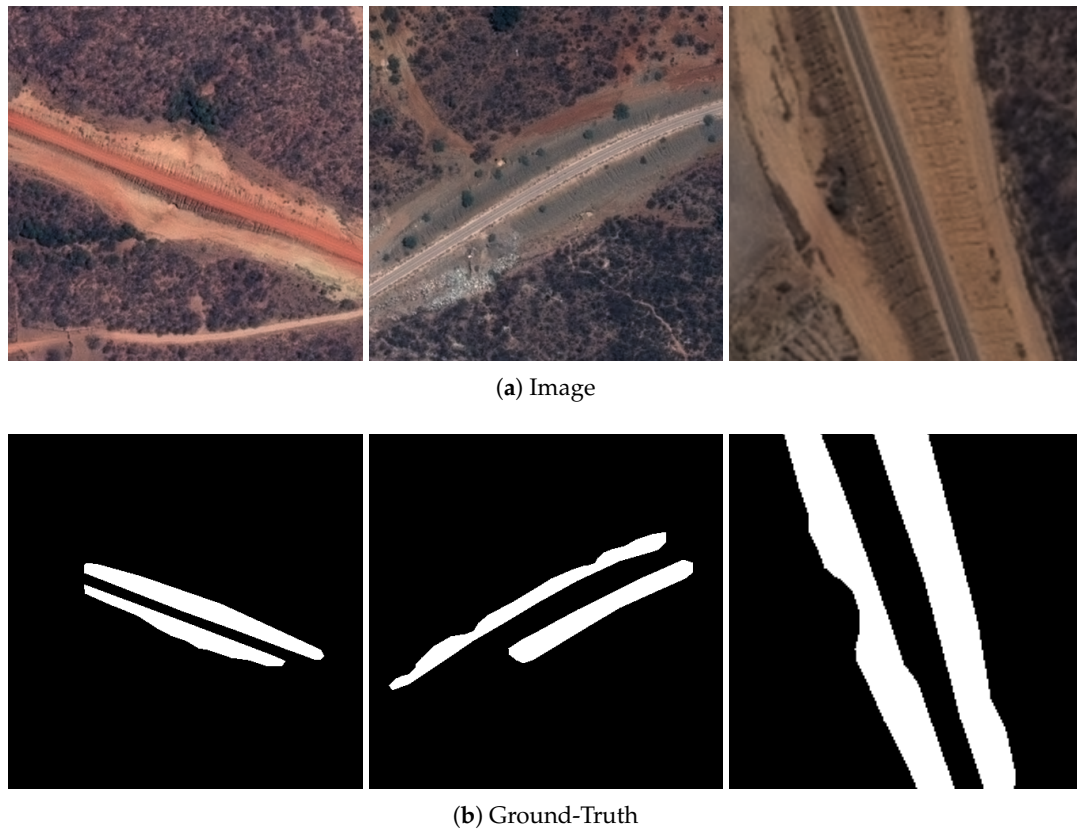


Figure 10. Three images of *Railway Erosion Dataset* and their respective ground-truths. The first two images are in the original size (i.e., 448×448), while the last one is an upsampled version of the original image in order to allow a better view of erosion areas. Legend – White: Erosion areas. Black: Non-erosion areas/Background.

Each method tested in this work was trained considering the set of hyperparameters presented in Table 1. Except for the number of iterations and batch size, all other parameters were preserved from the original works published in the literature. In the tested models, the Learning Rate (LR), responsible for determining how much the network weights will change based on the current iteration, starts with a high value and is reduced during the training phase according to the LR Decay column of Table 1. Through Table 1, it is possible to observe that DeepLab [12] and Mask R-CNN [13] start the training process with a smaller value of learning rate (when compared to the other methods). This is due to the fact that such approaches use pre-trained networks (as explained in Section 4) while the other techniques were trained from scratch for the proposed dataset. Aside from this, note that the early stopping technique [4] was employed in all trained approaches.

Table 1. Hyperparameters employed in each tested technique.

Method	Learning Rate (LR)	Weight Decay	LR Decay (decay/steps)	Batch Size	Iterations
Pixelwise [7]	0.01	0.0005	0.5/50,000	256	10,000
FCN [8]	0.001	0.0001	0.5/50,000	5	70,000
DeconvNet [9,10]	0.001	0.0001	0.5/50,000	6	70,000
DeepLabV3+ [12]	0.0001	0.00004	0.1/15,000	20	60,000
Dynamic Dilated ConvNet [5]	0.01	0.001	0.5/50,000	32	150,000
Mask R-CNN [13]	0.0001	0.0001	0.1/400,000	2	600,000

5.3. Software and Hardware Setup

All deep learning-based models exploited in this work were implemented using TensorFlow [51], a Python framework conceived to allow efficient exploitation of deep learning with Graphics Processing Units (GPUs). All experiments were performed on a 64-bit Intel i7 5930K machine with 3.5 GHz of clock, 64 GB of RAM memory. Two GeForce GTX TITAN Xp with 12 GB of memory, under an 10.0 CUDA version, were employed in this work. Note, however, that each GPU was used independently and that all networks exploited here can be trained using only one GPU. Ubuntu 18.04.3 LTS was used as the operating system, with ArcGIS 10.5 and Python version 3.5.2.

6. Results

A systematic set of experiments was conducted to: (i) evaluate the effectiveness and robustness of the aforementioned approaches, and (ii) define the best technique to be integrated into the proposed framework and plugin.

The obtained results of the evaluated methods are presented in Figure 11. Moreover, Table 2 relates the obtained results with the training time (in hours per fold) for each evaluated technique. In addition, a visual comparison of the outcomes generated by each approach is presented in Figure 12.

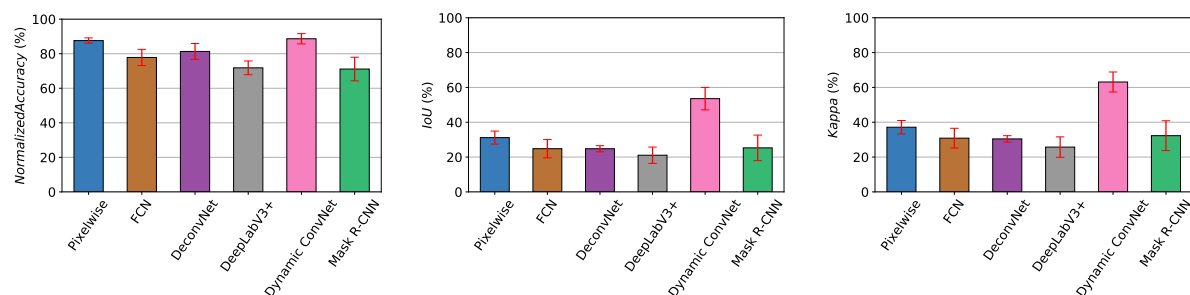


Figure 11. Graphical analysis of the evaluated methods for normalized accuracy, Intersection over Union (IoU), and Kappa metrics.

Table 2. Results and training time of the evaluated approaches for the Railway Erosion dataset.

Method	Normalized Accuracy (%)	IoU (%)	Kappa (%)	Training Time (Hours per Fold)
Pixelwise [7]	87.59 ± 1.56	31.18 ± 3.74	37.15 ± 3.85	15
FCN [8]	77.83 ± 4.68	24.82 ± 5.25	30.86 ± 5.66	24
DeconvNet [9,10]	81.31 ± 4.60	24.79 ± 1.76	30.43 ± 1.86	20
DeepLabV3+ [36]	71.82 ± 3.98	21.06 ± 4.69	25.74 ± 5.87	12
Dynamic Dilated ConvNet [5]	88.65 ± 2.98	53.55 ± 6.45	63.11 ± 5.75	150
Mask R-CNN [13]	71.12 ± 6.84	25.29 ± 7.34	32.29 ± 8.57	15

Overall, all tested methods, except the Dynamic Dilated ConvNet [5], produced very similar results for all evaluated metrics while having a comparable training time. The DDCN [5] also yielded similar results, in terms of normalized accuracy, when compared to all other methods, as seen in Table 2. However, considering the other metrics, it is possible to note that the DDCN [5] achieved the best results outperforming all other techniques that had very similar performance as previously stated, as seen in Figure 11.

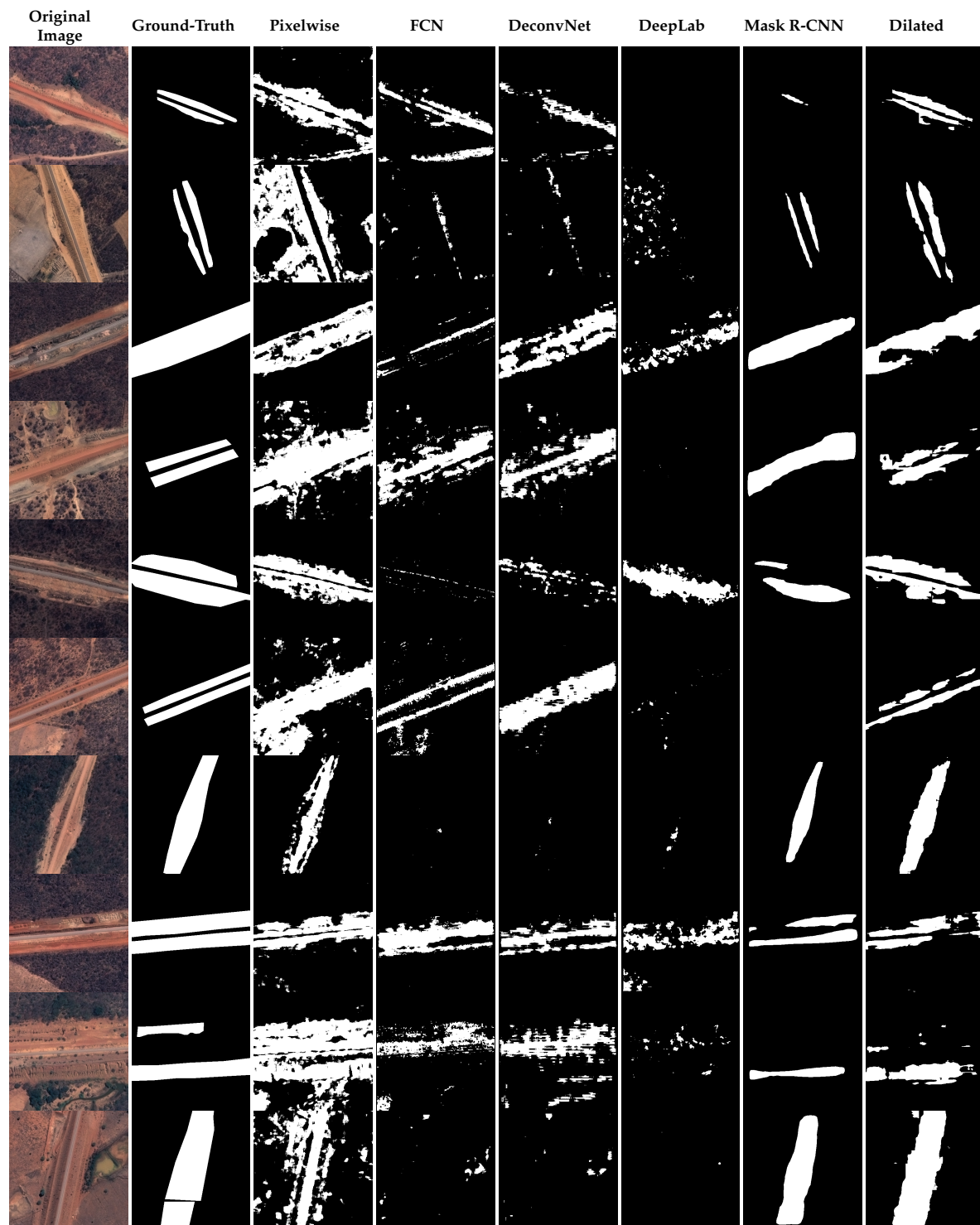


Figure 12. Images of the Railway Erosion dataset, their respective ground-truths, and the prediction maps generated by all evaluated algorithms. Legend—White: Erosion areas. Black: Non-erosion areas/Background.

The explanation for this outcome is two-fold. The first one would be due to the training strategy, employed by the Dynamic Dilated ConvNet [5] that allows the model to easily aggregate multi-scale information. The second one would be that such dilated network never downsamples the input image, preserving its resolution during all processing. This allows the network to extract even more useful information from the data. The drawback of this method is training time (as can be seen in Table 2),

which is huge compared to the other evaluated methods. Again, this may be justified by the fact that the input image is not downsampled, resulting in a model that requires more time to converge as the processing time of each layer is proportional to the input data resolution [45].

In addition to the previous results, Figure 12 presents several visual outcomes generated by all experimented approaches. As expected, the best visual outcomes are produced by the DDCN [5]. Again, this may be explained by the dynamic multi-scale training strategy (employed in this technique) and the preservation of the input resolution through the network. Hence, despite the disadvantages and based on (both quantitative and qualitative) obtained results, the Dynamic Dilated ConvNet [5] was the technique selected, implemented, and encapsulated into the aforementioned ArcGIS plugin.

7. Discussion

As presented in the previous section, the DDCN [5] technique yielded the best outcomes and was integrated into the final version of the proposed tool and ArcGIS plugin. However, as aforementioned, the main disadvantage of such an approach is its demanding training time. Although this might seem to be a problem, note that the computational time of this technique is expensive only during the training phase. The inference (or testing) time of this method is small since no optimization is carried out in the model anymore [4], i.e., no back-propagation calculations are computed and only the feed-forward step is performed. Therefore, once the model is trained, it can be used repeatedly to identify erosion areas in different scenarios without consuming too much time.

Considering the compulsory computation time of the inference procedure, we argue that the proposed framework is useful and can be used to decrease both the time and resources required to perform erosion identification. Comparing to the current approaches, such as the *in loco* monitoring and the manual process, the inference time and training cost is negligible. This is because *in loco* monitoring requires at least one specialist to visit each investigated location to assess the erosion level while the manual investigation requires specialists to visually analyze large-scale remote sensing datasets in order to recognize erosion areas in different scenarios and cases. Since both situations require at least one specialist to manually identify erosion areas, the actual computational cost of training and using a Dynamic Dilated ConvNet [5] model outweighs the time and economic cost involved in those situations.

Another important aspect to mention is that, in this work, an ArcGIS plugin that encapsulates the proposed framework, was implemented, opening opportunities for a simplified use of this powerful deep learning-based technique by non-programmers and users. In fact, we expect that any end-user, with minimal knowledge of the ArcGIS software and no programming skills, is able to exploit this plugin and speed up their work.

Overall, we believe that the solution and plugin proposed in this work may help several distinct public and private organizations. An example of usefulness of this work would be to assist control agencies in auditing public works. These auditing process is, usually, realized through *in loco* monitoring, or visual analysis of remote sensing imagery, performed by agents [52,53]. In cases like this, automatic machine learning-based methods can be used as an acceptable solution to speed up the audit process.

Aside from this, note that no comparison with baselines from the literature was performed. This is because, although there are several works performing erosion identification [3,14–23], most of them are based on the combination of multiple information (such as Digital Terrain Models [15], Digital Elevation Models [3], and temporal information [16,22,23]). Due to this, such existing approaches can be seen as different from the proposed tool, which is exclusively based on remote sensing images. Therefore, given this essential difference, we argue that a comparison between those works from the literature and the proposed one would not be totally fair.

8. Conclusions

In this paper, we proposed a novel framework, based on Convolutional Networks, to perform erosion identification (i.e., segmentation and classification) in railway lines using high-resolution aerial image sets. In order to define the best deep learning-based approach to be integrated into the proposed framework, we evaluated, using a proposed Railway Erosion dataset, six existing techniques, including: (i) Pixelwise [7], (ii) Fully Convolutional Network (FCN) [8], (iii) Deconvolution network [9–11], (iv) DeepLabV3+ [12], (v) DDCN [5], and (vi) Mask R-CNN [13].

Experiments made on this dataset have shown that the Dynamic Dilated ConvNet [5] produces the best results (in terms of IoU and Kappa) for the proposed dataset. Based on obtained results, the DDCN [5] was the technique selected and implemented in proposed framework, which was then converted to a new plugin for the software ArcGIS.

The presented work opens opportunities towards a simplified use of deep learning methods for better identification of erosion in railway lines, an important application to solve potentially impactful problems, such as economic losses and human casualties. Furthermore, we argue that the proposed framework (and plugin) is able to significantly decrease the time invested in identifying erosion in large areas, such as the *Transnordestina Railway*. Even if the model's training time is substantial, the inference time is small and the effort required to analyze and identify erosion areas in the proposed dataset is still smaller than an *in loco* monitoring or a manual observation of the images.

As future work, we plan to increase the proposed dataset by including images from railway lines of other parts of the world, making it more robust and representative. We also intend to experiment and integrate more deep learning-based algorithms into the proposed tool and plugin. Finally, we also plan to evaluate unsupervised deep learning methods for erosion identification, comparing such approaches with supervised ones.

Author Contributions: The research topic was coordinated by J.A.S. and designed by K.N.; K.N., G.L.S.M., and P.H.T.G. conducted the implementation and execution of the experiments. R.B. was responsible for the data acquisition and pre-processing. K.N. wrote the first version of manuscript, assisted by G.L.S.M., and P.H.T.G.; C.C.V.d.S., R.B., and J.A.d.S. were responsible for the final revision of the manuscript. All authors have read and agreed to the published version of the manuscript.

Funding: This research was partially supported by the Conselho Nacional de Desenvolvimento Científico e Tecnológico—CNPq (Grant No. 311395/2018-0 and No. 424700/2018-2), the Coordenação de Aperfeiçoamento de Pessoal de Nível Superior—Brasil (CAPES), and the Fundação de Amparo à Pesquisa do Estado de Minas Gerais—Fapemig (APQ-00449-17).

Acknowledgments: We thank the German Agency for International Cooperation (GIZ) and for the Brazilian Federal Court of Accounts (TCU) for the partnership. The authors gratefully acknowledge the support of the NVIDIA Corporation with the donation of the GeForce GTX TITAN X GPU to the PATREO Laboratory that were used in this work.

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

References

1. Eswaran, H.; Lal, R.; Reich, P.F. Land degradation: An overview. *Responses Land Degrad.* **2001**, *1*, 20–35.
2. Castillo, C.; Gómez, J. A century of gully erosion research: Urgency, complexity and study approaches. *Earth-Sci. Rev.* **2016**, *160*, 300–319. [[CrossRef](#)]
3. Liu, K.; Ding, H.; Tang, G.; Na, J.; Huang, X.; Xue, Z.; Yang, X.; Li, F. Detection of catchment-scale gully-affected areas using unmanned aerial vehicle (UAV) on the Chinese Loess Plateau. *ISPRS Int. J. Geo-Inf.* **2016**, *5*, 238. [[CrossRef](#)]
4. Goodfellow, I.; Bengio, Y.; Courville, A.; Bengio, Y. *Deep Learning*; MIT Press: Cambridge, MA, USA, 2016; Volume 1.
5. Nogueira, K.; Dalla Mura, M.; Chanussot, J.; Schwartz, W.R.; Dos Santos, J.A. Dynamic Multicontext Segmentation of Remote Sensing Images Based on Convolutional Networks. *IEEE Trans. Geosci. Remote Sens.* **2019**, *57*, 7503–7520. [[CrossRef](#)]

6. Nogueira, K.; Penatti, O.A.; dos Santos, J.A. Towards better exploiting convolutional neural networks for remote sensing scene classification. *Pattern Recognit.* **2017**, *61*, 539–556. [[CrossRef](#)]
7. Nogueira, K.; Dalla Mura, M.; Chanussot, J.; Schwartz, W.R.; dos Santos, J.A. Learning to semantically segment high-resolution remote sensing images. In Proceedings of the International Conference on Pattern Recognition, Cancun, Mexico, 4–8 December 2016; pp. 3566–3571.
8. Long, J.; Shelhamer, E.; Darrell, T. Fully convolutional networks for semantic segmentation. In Proceedings of the Computer Vision and Pattern Recognition, Boston, MA, USA, 7–12 June 2015; pp. 3431–3440.
9. Noh, H.; Hong, S.; Han, B. Learning deconvolution network for semantic segmentation. In Proceedings of the International Conference on Computer Vision, Las Condes, Chile, 13–16 December 2015; pp. 1520–1528.
10. Badrinarayanan, V.; Kendall, A.; Cipolla, R. Segnet: A deep convolutional encoder-decoder architecture for image segmentation. *IEEE Trans. Pattern Anal. Mach. Intell.* **2017**, *39*, 2481–2495. [[CrossRef](#)]
11. Yang, W.; Zhou, Q.; Lu, J.; Wu, X.; Zhang, S.; Latecki, L.J. Dense deconvolutional network for semantic segmentation. In proceedings of the International Conference on Image Processing, Athens, Greece, 7–10 October 2018; pp. 1573–1577.
12. Chen, L.C.; Zhu, Y.; Papandreou, G.; Schroff, F.; Adam, H. Encoder-Decoder with Atrous Separable Convolution for Semantic Image Segmentation. In Proceedings of the European Conference on Computer Vision, Munich, Germany, 8–14 September 2018; pp. 801–818.
13. He, K.; Gkioxari, G.; Dollár, P.; Girshick, R. Mask r-cnn. In Proceedings of the International Conference on Computer Vision, Veneza, Itália, 22–29 October 2017; pp. 2961–2969.
14. Benzer, N. Using the geographical information system and remote sensing techniques for soil erosion assessment. *Pol. J. Environ. Stud.* **2010**, *19*, 881–886.
15. d’Oleire Oltmanns, S.; Marzolf, I.; Peter, K.; Ries, J. Unmanned aerial vehicle (UAV) for monitoring soil erosion in Morocco. *Remote Sens.* **2012**, *4*, 3390–3416. [[CrossRef](#)]
16. Eltner, A.; Mulsow, C.; Maas, H. Quantitative measurement of soil erosion from TLS and UAV data. *Int. Arch. Photogramm. Remote Sens. Spat. Inf. Sci.* **2013**, *40*, 4–6. [[CrossRef](#)]
17. Žižala, D.; Zádorová, T.; Kapička, J. Assessment of soil degradation by erosion based on analysis of soil properties using aerial hyperspectral images and ancillary data, Czech Republic. *Remote Sens.* **2017**, *9*, 28. [[CrossRef](#)]
18. Arif, N.; Danoedoro, P.; Hartono, B. Analysis of Artificial Neural Network in Erosion Modeling: A Case Study of Serang Watershed. *IOP Conf. Ser. Earth Environ. Sci.* **2017**, *98*, 012027. [[CrossRef](#)]
19. Krenz, J.; Greenwood, P.; Kuhn, N.J. Soil Degradation Mapping in Drylands Using Unmanned Aerial Vehicle (UAV) Data. *Soil Syst.* **2019**, *3*, 33. [[CrossRef](#)]
20. Xu, H.; Hu, X.; Guan, H.; Zhang, B.; Wang, M.; Chen, S.; Chen, M. A Remote Sensing Based Method to Detect Soil Erosion in Forests. *Remote Sens.* **2019**, *11*, 513. [[CrossRef](#)]
21. Vågen, T.G.; Winowiecki, L.A. Predicting the Spatial Distribution and Severity of Soil Erosion in the Global Tropics using Satellite Remote Sensing. *Remote Sens.* **2019**, *11*, 1800. [[CrossRef](#)]
22. Gianinetto, M.; Aiello, M.; Polinelli, F.; Frassy, F.; Rulli, M.C.; Ravazzani, G.; Bocchiola, D.; Chiarelli, D.D.; Soncini, A.; Vezzoli, R. D-RUSLE: A dynamic model to estimate potential soil erosion with satellite time series in the Italian Alps. *Eur. J. Remote. Sens.* **2019**, *52*, 34–53. [[CrossRef](#)]
23. Peponi, A.; Morgado, P.; Trindade, J. Combining Artificial Neural Networks and GIS Fundamentals for Coastal Erosion Prediction Modeling. *Sustainability* **2019**, *11*, 975. [[CrossRef](#)]
24. Renard, K.G.; Foster, G.R.; Weesies, G.; McCool, D.; Yoder, D. *Predicting Soil Erosion by Water: A Guide to Conservation Planning with the Revised Universal Soil Loss Equation (RUSLE)*; United States Department of Agriculture: Washington, DC, USA, 1997; Volume 703.
25. Baatz, M.; Schape, A. Multiresolution Segmentation: An Optimization Approach for High Quality Multi-Scale Image Segmentation. In Proceedings of the XII Angewandte Geographische Informations-Verarbeitung, Salzburg, Germany, July 2000.
26. Längkvist, M.; Kiselev, A.; Alirezaie, M.; Loutfi, A. Classification and segmentation of satellite orthoimagery using convolutional neural networks. *Remote Sens.* **2016**, *8*, 329. [[CrossRef](#)]
27. Fu, G.; Liu, C.; Zhou, R.; Sun, T.; Zhang, Q. Classification for high resolution remote sensing imagery using a fully convolutional network. *Remote Sens.* **2017**, *9*, 498. [[CrossRef](#)]

28. Nogueira, K.; Fadel, S.G.; Dourado, Í.C.; Werneck, R.d.O.; Muñoz, J.A.; Penatti, O.A.; Calumby, R.T.; Li, L.T.; dos Santos, J.A.; Torres, R.d.S. Exploiting ConvNet diversity for flooding identification. *IEEE Geosci. Remote Sens. Lett.* **2018**, *15*, 1446–1450. [\[CrossRef\]](#)
29. Marmanis, D.; Schindler, K.; Wegner, J.D.; Galliani, S.; Datcu, M.; Stilla, U. Classification with an edge: Improving semantic image segmentation with boundary detection. *ISPRS J. Photogramm. Remote Sens.* **2018**, *135*, 158–172. [\[CrossRef\]](#)
30. Nogueira, K.; dos Santos, J.A.; Menini, N.; Silva, T.S.; Morellato, L.P.C.; Torres, R.d.S. Spatio-Temporal Vegetation Pixel Classification By Using Convolutional Networks. *IEEE Geosci. Remote Sens. Lett.* **2019**, *16*, 1665–1669. [\[CrossRef\]](#)
31. Bakator, M.; Radosav, D. Deep learning and medical diagnosis: A review of literature. *Multimodal Technol. Interact.* **2018**, *2*, 47. [\[CrossRef\]](#)
32. Al-Bander, B.; Williams, B.; Al-Nuaimy, W.; Al-Tae, M.; Pratt, H.; Zheng, Y. Dense fully convolutional segmentation of the optic disc and cup in colour fundus for glaucoma diagnosis. *Symmetry* **2018**, *10*, 87. [\[CrossRef\]](#)
33. Ma, B.; Ban, X.; Huang, H.; Chen, Y.; Liu, W.; Zhi, Y. Deep learning-based image segmentation for al-la alloy microscopic images. *Symmetry* **2018**, *10*, 107. [\[CrossRef\]](#)
34. Hesamian, M.H.; Jia, W.; He, X.; Kennedy, P. Deep Learning Techniques for Medical Image Segmentation: Achievements and Challenges. *J. Digit. Imaging* **2019**, *32*, 582–596. [\[CrossRef\]](#)
35. Zhou, T.; Ruan, S.; Canu, S. A review: Deep learning for medical image segmentation using multi-modality fusion. *Array* **2019**, *3*, 100004. [\[CrossRef\]](#)
36. Sandler, M.; Howard, A.; Zhu, M.; Zhmoginov, A.; Chen, L.C. MobileNetV2: Inverted Residuals and Linear Bottlenecks. In Proceedings of the Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–22 June 2018; pp. 4510–4520.
37. Zeiler, M.D.; Fergus, R. Visualizing and understanding convolutional networks. In Proceedings of the European Conference on Computer Vision, Zurich, Switzerland, 6–12 September 2014; pp. 818–833.
38. Chen, L.C.; Papandreou, G.; Kokkinos, I.; Murphy, K.; Yuille, A.L. Semantic image segmentation with deep convolutional nets and fully connected crfs. *arXiv* **2014**, arXiv:1412.7062.
39. Sherrah, J. Fully convolutional networks for dense semantic labelling of high-resolution aerial imagery. *arXiv* **2016**, arXiv:1606.02585.
40. Chen, L.C.; Papandreou, G.; Kokkinos, I.; Murphy, K.; Yuille, A.L. Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs. *IEEE Trans. Pattern Anal. Mach. Intell.* **2017**, *40*, 834–848. [\[CrossRef\]](#)
41. Chen, L.C.; Papandreou, G.; Schroff, F.; Adam, H. Rethinking atrous convolution for semantic image segmentation. *arXiv* **2017**, arXiv:1706.05587.
42. Yu, F.; Koltun, V. Multi-scale context aggregation by dilated convolutions. *arXiv* **2015**, arXiv:1511.07122.
43. Chollet, F. Xception: Deep learning with depthwise separable convolutions. In Proceedings of the Computer Vision and Pattern Recognition, Honolulu, HI, USA, 21–26 July 2017; pp. 1251–1258.
44. Everingham, M.; Eslami, S.A.; Van Gool, L.; Williams, C.K.; Winn, J.; Zisserman, A. The pascal visual object classes challenge: A retrospective. *Int. J. Comput. Vis.* **2015**, *111*, 98–136. [\[CrossRef\]](#)
45. He, K.; Sun, J. Convolutional neural networks at constrained time cost. In Proceedings of the Computer Vision and Pattern Recognition, Boston, MA, USA, 7–12 June 2015; pp. 5353–5360.
46. Ren, S.; He, K.; Girshick, R.; Sun, J. Faster r-cnn: Towards real-time object detection with region proposal networks. In Proceedings of the Advances in Neural Information Processing Systems, Montreal, QC, Canada, 7–12 December 2015; pp. 91–99.
47. He, K.; Zhang, X.; Ren, S.; Sun, J. Deep residual learning for image recognition. In Proceedings of the Computer Vision and Pattern Recognition, Alsace, France, 13–15 July 2016; pp. 770–778.
48. Deng, J.; Dong, W.; Socher, R.; Li, L.J.; Li, K.; Fei-Fei, L. Imagenet: A large-scale hierarchical image database. In Proceedings of the Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 13–16 July 2009; pp. 248–255.
49. Csurka, G.; Larlus, D.; Perronnin, F.; Meylan, F. What is a good evaluation measure for semantic segmentation? In Proceedings of the British Machine Vision Conference, Bristol, UK, 9–13 September 2013; Volume 27, p. 2013.

50. Ferri, C.; Hernández-Orallo, J.; Modroiu, R. An experimental comparison of performance measures for classification. *Pattern Recognit. Lett.* **2009**, *30*, 27–38. [[CrossRef](#)]
51. Abadi, M.; Barham, P.; Chen, J.; Chen, Z.; Davis, A.; Dean, J.; Devin, M.; Ghemawat, S.; Irving, G.; Isard, M.; others. Tensorflow: A system for large-scale machine learning. In Proceedings of the 12th {USENIX} Symposium on Operating Systems Design and Implementation ({OSDI} 16), Savannah, GA, USA, 2–4 November 2016; pp. 265–283.
52. Sayed, M.; Mhaske, S. GIS based Road Safety Audit. *Int. J. Sci. Eng. Res. (IJSER)* **2013**, *1*, 21–23.
53. De Carvalho, O.A.; Trancoso, R.A.; Guimarães, R.F. The potential of remote sensing data in public works audit. *RTCU* **2016**. Available online: <http://revista.tcu.gov.br/ojs/index.php/RTCU/issue/view/68/showToc> (accessed on 19 July 2017).



© 2020 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<http://creativecommons.org/licenses/by/4.0/>).