



# **Automatic forest road extraction from LiDAR data using convolutional neural networks**

Paul Georges, Phuc Ngo, Philippe Even

## **► To cite this version:**

Paul Georges, Phuc Ngo, Philippe Even. Automatic forest road extraction from LiDAR data using convolutional neural networks. Reproducible Research in Pattern Recognition (RRPR), Aug 2022, Montreal, Canada. pp.91-107, <10.1007/978-3-031-40773-4\_8>. <hal-03886785>

**HAL Id: hal-03886785**

**<https://hal.science/hal-03886785v1>**

Submitted on 25 Aug 2023

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.



HAL Authorization

# Automatic forest road extraction from LiDAR data using convolutional neural networks

Paul Georges<sup>1</sup>, Phuc Ngo<sup>1</sup> and Philippe Even<sup>1</sup>

Université de Lorraine, CNRS, LORIA, 54000 Nancy, France  
(hoai-diem-phuc.ngo,philippe.even)@loria.fr

**Abstract.** Accurate location of access roads is important for forest management, in particular in mountain areas. In this paper, we are interested in their detection from LiDAR data using deep learning approaches. For this, we use images computed from an interpolated surface, called *digital terrain model* (DTM), of the 3D point cloud. In order to train and validate the neural network models, two ground truth datasets associated to DTM images are considered: (1) manual digitization of the road centerlines and (2) automatic extraction followed by supervised completion using two softwares based on discrete geometry tools. The trained network models are then evaluated over a test dataset using standard measures such as precision, recall, F-measure and prediction time.

**Keywords:** Road detection, mountainous area, LiDAR images, CNN

## 1 Introduction

Forest road location and characterization are important information used for various purposes in forest management and such activities as wood harvesting. Their maintenance is also of great concern for planning team interventions on forest fires, in particular in mountain areas where accessibility may be quite difficult. In this context, airborne laser scanning, also called LiDAR (Light Detection And Ranging), is of great help to survey forested mountain areas. It is a 3D acquisition technique based on the emission of a laser beam swept over the measured scene and on the detection of reflected signal from the surface. In forested terrains, the received signal is composed of multiple echoes corresponding to the successive hit obstacles, from the forest canopy, down to lower vegetation levels and finally to the ground itself. From lower cloud points, classified as *ground points*, an interpolated surface, called *digital terrain model* (DTM), is computed using optimization techniques. An example of DTM image is given in *Fig. 1 a*.

Numerous studies have been proposed in the literature for road extraction from LiDAR data. Most of them deal with urban and peri-urban areas, for instance [1,12]. They are not well suited to rural context. Large standardized logging roads (see *Fig. 1 b*) are easily detected, but the sole topographic information is not sufficient to distinguish the pathway from bare earth borders

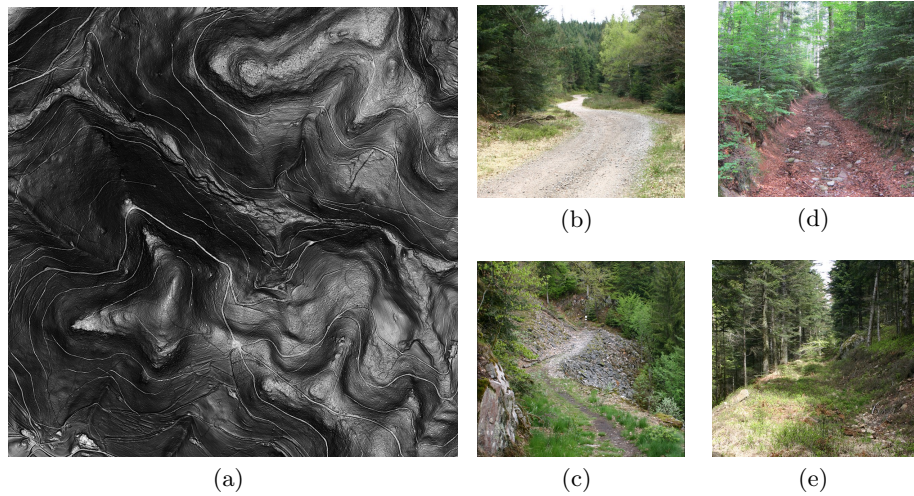


Fig. 1: (a) A shaded digital terrain model (DTM) image of size  $4000 \times 4000$  pixels and  $0.25 \text{ m}^2/\text{pixel}$  resolution, that corresponds to a  $4 \text{ km}^2$  area. (b–e) Different forest road types in mountainous context: (b) logging road with large vegetated shoulders at the same height, (c) shrunk road crossing natural screes, (d) eroded hollow road, and (e) unused road left to growing vegetation.

at the same level, that are used for log stocking. They are generally well documented and not of major interest in our study. Smaller roads are more difficult to process as they show quite irregular surfaces and strong width variations along their layout. This may come from local terrain constraints (see *Fig. 1 c*), natural erosion (see *Fig. 1 d*), irregular maintenance or even complete relinquishment when they are not anymore used (see *Fig. 1 e*). Some attempts have been done to use the DTM to detect main roads on large-scale LiDAR [10,16,22,24]. Fewer works suggested to process raw data in complement to DTM analysis [1,2,12]. In particular, signal intensity was used to recognize road surface response. However, these approaches depend strongly on local terrain features, and parameters are difficult to set in practice. Raw altimetric information could also help to better discriminate roads, but its processing is generally considered as complicated.

Recently, a new framework, based on efficient discrete geometry and mathematical morphology tools, has been proposed in [8] for extracting automatically forest roads from LiDAR data. It is composed of two steps. First, the DTM image is analyzed to find relevant locations for detecting roads, then for each of these selected seeds, road sections are extracted using only raw LiDAR ground points. By processing ground points, the detection is more aware of the heterogeneous point distribution in the raw data. This helps to overcome the limits of DTM interpolation. The extraction framework was successfully tested on a large-scale LiDAR dataset. However, as mentioned in [8], it may not provide a good detection for complex road sections such as tight curves and intersections.

Convolutional neural networks (CNNs) based on deep learning technology are widely used in many different applications of image processing and analysis including semantic image segmentation. Due to their learning capabilities from training data, deep learning methods have performed remarkably well on many image analysis tasks and they lead to very successful results compared with traditional methods. Recently, they have been applied in road extraction from airborne LiDAR data [16,22]. In [22], a pixel-to-pixel architecture based on fully convolutional neural network is used to perform automatic mapping of small roads. This network provides good results for large-scale data. However, it may be bad at making local adaptations as mentioned by the authors. Higher performance could be achieved, but at the price of larger complexity and computational cost. In [16], a Dense Dilated Convolutions Merging Network (DDCM-Net) is proposed for multi-class segmentation of forest roads in the purpose of mapping road networks. The proposed architecture relies on multiple dilated convolutions merged with various dilation rates. It allows the network to learn more robust feature representation with densely linked dilated convolutions and to recognize effectively multi-scale and complex-shaped roads. In [17], this work was extended to a more general segmentation task using grouped convolutions and strided convolutions, in order to enhance the discrimination of small objects for a complete land cover classification task.

Inspired by the works [7,8,16], the current study aims at developing a fast, accurate and operationally simple deep-learning-based method, which considers information obtained from shaded DTM maps to detect the forest road in mountainous areas. More precisely, we use two softwares based on discrete geometry tools: ILSD (Interactive Linear Structure Detector) [7] and AMREL (Automatic Mountain Road Extraction from LiDAR data) [8] to generate the training dataset of forest roads from DTM images and LiDAR data. These data are fed to a convolutional neural network. In particular, we consider the binary branch of DDCM-Net [16] designed for a road segmentation task, and here adapted to the detection of narrow forest roads. Actually, it is quite difficult to segmentate these changing objects on a geometrical basis, in particular to provide reliable ground truths. The trained network models are then evaluated over a test dataset using the standard measure (precision, recall, F-measure). Another dataset, with manual annotations of forest roads represented by their centerline, is used to evaluate the improvement and the efficiency of the neural network in comparison to AMREL results [8] for forest road detection. The implementation of the neural network architecture, the trained models and execution procedure for road predictions are available at: <https://github.com/paulgeorges1998/Light-DDCM-Net>.

The rest of the paper is organized as follows: *Sec. 2* describes our problem of forest road extraction and recalls the previous works related to this problem as well as the DDCM-Net architecture used in this work. *Sec. 3* explains the experimental setup: dataset and network training, and *Sec. 4* presents the experimental results. Finally, *Sec. 5* gives a conclusion and draws some perspectives.

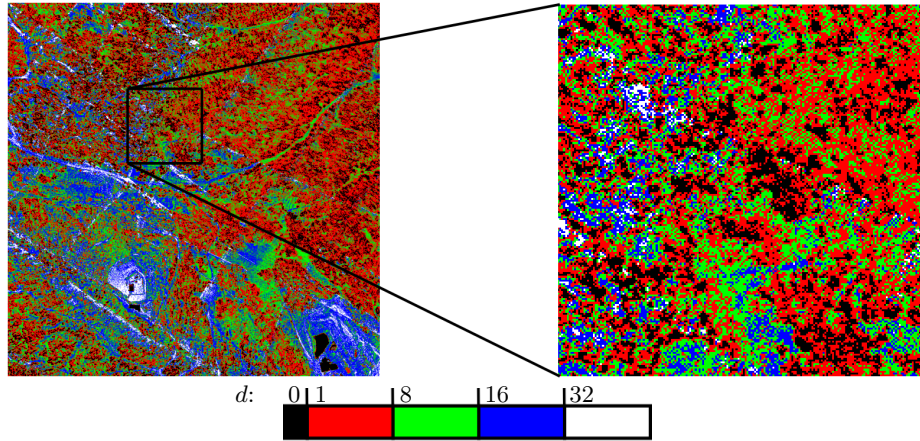


Fig. 2: Detail of the ground point density map; the color corresponds to the number  $d$  of points per pixel ( $1 \text{ m}^2$ ).

## 2 Method

### 2.1 Problem statement

The present work addresses the problem of automatic extraction of forest roads from the grayscale images of digital terrain model (DTM) associated to LiDAR ground point cloud. The DTM is encoded as a normal vector map obtained by derivation of the original height map. To visualize a DTM, hill-shading is a widespread technique based on controllable directional light sources. For detection purpose, we rather use slope-shading which can be seen as a lighting by a zenital source obtained by the normal vector z-component. As pointed out in [24], slope-shading ensures a good contrast between low gradient road surface and steep adjacent road cuts. In mountainous context, the interest objects – forest roads – correspond to *mostly flat zones* compared to the surroundings and this distinguishes them from the background. The slope-shaded DTM is a good representation allowing to enhance the flat zones. This is very beneficial for a learning process with neural networks to detect the forest roads.

Contrarily to road detection in urban and peri-urban areas using LiDAR data in which road characteristics are quite regular and well-contrasted, the case of forest roads is more challenging because of their wide range of shape and geometric features, and moreover, the large variations of the ground point density. Actually, dense vegetation impedes the laser beam from reaching the soil, and thus produces a heterogeneous distribution of the ground points as illustrated in *Fig. 2*. Such local lack of points makes the road detection task more difficult. Furthermore, sparse data may cause large approximations in the delivered DTM.

The purpose of the present study is to investigate the performance of a simple *deep-learning-based algorithm* combined with available LiDAR and image

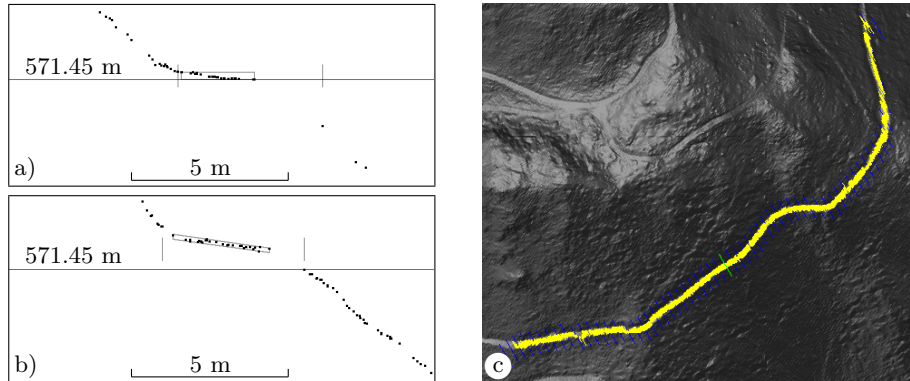


Fig. 3: A forest road extraction: a) altimetric profile collected at start scan (matched plateau enclosed, inaccurate right border location because of point lack); b) altimetric profile collected at scan 20 (tilted plateau); c) detected road over the DTM view (one scan on ten is displayed, manual seed in green, next scans in blue).

processing tools to automatically detect forest roads in mountainous areas using shaded DTM images. More specifically, this problem can be seen as a binary segmentation with DTM images as input, and a binary image as output with white pixels indicating forest roads and black pixels for background, *i.e.*, not forest road. To that end, we selected a network architecture based on dilated convolutions at different rates [26]. Considering the complexity of goal objects, such convolutions will allow the network to have very large receptive fields and to learn scale-invariant features, and therefore recognize effectively multi-scale and complex-shaped roads with similar textures and intensities from input images.

## 2.2 Previous approach to forest road extraction

This work follows a former project on linear structure extraction, that resulted in two open-source softwares: ILSD [6] and AMREL [5].

ILSD allows the interactive extraction of linear structures, such as ridges, holloways or forest roads, from LiDAR raw data, *i.e.*, the set of 3D points classified as “ground” [4]. It relies on a scanline approach, where the user draws a stroke across a visible structure in the DTM view. Ground points lying in this manual seed are collected to get an altimetric profile, matched to a model of the structure cross profile (see *Fig. 3 a*). In the case of forest roads, the selected cross profile is approximated by a nearly horizontal plateau bounded by slope sides. This model fits well to mountainous context. In case of success, the structure is extended on both sides of the manual seed using adjacent scans and collected profiles (see *Fig. 3 b*). Spatial consistency is checked between successive altimetric profiles. All the processing is based on discrete geometry tools: blurred segments [3] or adaptive directional scans [9]. A several hundred meters long section can be extracted in a fraction of a second (see *Fig. 3 c*). Details of the extraction framework are provided in [7].

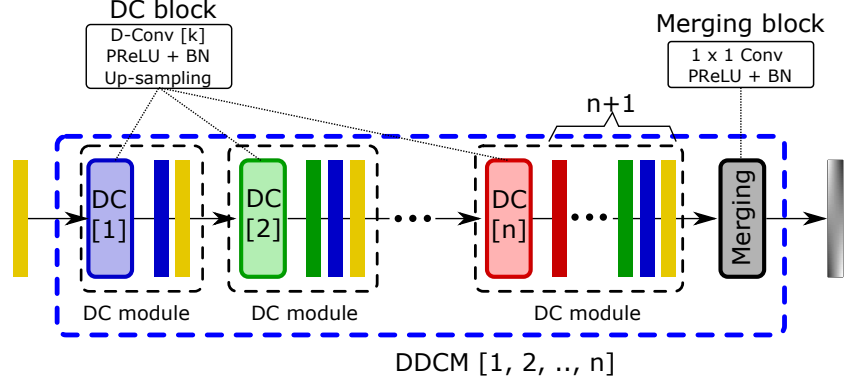


Fig. 4: A DDCM module is composed of  $n$  dilated convolution (DC) blocks of various dilation rates  $k \in \{1, 2, \dots, n\}$  and a merging block. Each DC block is composed of a dilated convolution (D-Conv) of rate  $k$ , followed by a parametric rectified linear unit (PReLU) activation, batch normalization (BN) and bilinear up-sampling. The output of DC block is then concatenated with its input together to feed the next layer. The merging block is composed of a  $1 \times 1$  convolution (Conv) with PReLU and BN in order to efficiently combine all stacked features generated by intermediate DC blocks.

AMREL is focused on the automatic extraction of forest roads from LiDAR data [8]. Here, input seeds are automatically selected by processing slope-shaded views of the DTM. First a mathematical morphology operator, RORPO [19], is applied to enhance elongated shapes. Then straight edges are detected using FBSD [9], and seeds are arranged across the longest ones at regular intervals. Finally, each seed is processed by the scanline approach to obtain a collection of road sections. Performance evaluations showed a recall measure (ratio of correct detection area on whole detection area) of  $70 \pm 3 \%$ , and a precision measure (ratio of correct detection area on ground truth area) of  $81 \pm 7 \%$ . False detections most often occur at places with similar cross profiles to roads, for instance talwegs or cultivation terraces, and also in flat areas where the road section model is not well adapted. Undetected road sections correspond (i) to tight changes in direction where more seeds are needed to cope with the scanline approach directionality, (ii) to roads with large slope or irregular surface exceeding parameter thresholds, (iii) to areas with low point density.

Both tools can reliably be used to provide road ground truth maps. Their main drawback is the large amount of parameters to set. Most are directly connected to application needs, such as high or low bounds of road width, slope or tilt thresholds, ... But others take more time to set as they rather control internal details of the detection process.

### 2.3 Light DDCM-Net architecture

In [16], a dense dilated convolutions merging network (DDCM-Net) and a joint-task learning structure with an iterative-random-weighting strategy for the joint-



loss are proposed for a multi-class segmentation of forest roads from 2-band LiDAR images. In the proposed architecture, the DDCM-Net uses dilated convolutions [26] of stride 2 to learn features at varying dilation rates and merging by a concatenation of feature map at each layer with the feature maps from all previous layers. Such densely linked dilated convolutions and the fusion of feature maps is called a *DDCM module* and illustrated in *Fig. 4*. The DDCM modules allow the network to have very large receptive fields with just a few layers and to capture scale-invariant features of the detected objects.

The DDCM-Net architecture is composed of multiple DDCM modules with progressively increasing dilation rates to learn and to capture scale-invariant features of the detected objects. In [16], the DDCM-Net is integrated in a joint-task learning strategy, called an *end-to-end pipeline of the Joint-Task DDCM-Net*, to perform road segmentation and mapping tasks. In particular, the proposed architecture is composed of two parts: an encoder of low level features encodes multi-scale contextual information from the initial 2-band LiDAR images by a DDCM module with 6 different dilation rates (1, 2, 3, 5, 7, 9), and a decoder of high level features decodes highly abstract representations learned from the deep residual network (ResNet) pre-trained on ImageNet [21] by 2 DDCM modules, one with rates 1, 2, 3 and 4, the other with rate 1. The low-level and high-level feature maps by DDCMs are then fused together to infer pixel-wise full-class probabilities. The network outputs a multi-class segmentation which predicts what types of roads are in the input and a binary segmentation which locates all roads. More details of the Joint-Task DDCM-Net architecture are given in [16].

In this work, we are interested in the binary segmentation of forest road location from DTM images. For this purpose, we consider only the binary branch of the Joint-Task DDCM-Net in [16], called *L-DDCM-Net* (for Light DDCM-Net). The L-DDCM-Net architecture used in this work is illustrated in *Fig. 5* in which we add a convolution of two  $3 \times 3$  kernels and a concatenation with the input image, to create an image of 3 channels, followed by a batch normalization (BN) and a parametric rectified linear unit (PReLU) on top of the pre-trained ResNet [21]. In compatibility to Tensorflow, we remove the bilinear up-sampling in DDCM modules and use the dilated convolution of stride 1 with same padding instead without noticeable performance loss. For the loss function, we consider the standard binary cross entropy (BCE) loss [11]. More details about the training of L-DDCM-Net are given in *Sec. 3.2*.

### 3 Experimental setup

#### 3.1 Dataset

To our knowledge, no LiDAR test set of forest roads with ground truth is publicly available. This is certainly due to a large variability between different acquisition contexts and terrain configurations and maybe also to the huge data storage required. Many high resolution LiDAR campaigns are designed for archaeological prospections to reveal small topological details. In regards to sensitive aspects of cultural heritage material, access to this data is most often restricted. This



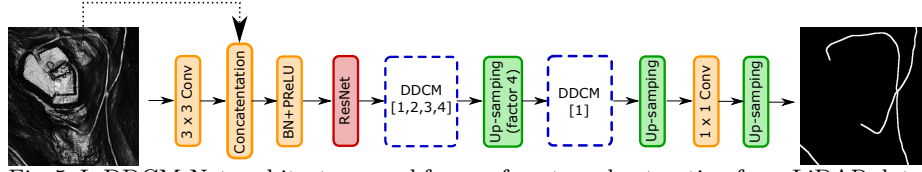


Fig. 5: L-DDCM-Net architecture used for our forest road extraction from LiDAR data. It is composed of a standard convolution, a concatenation with the input data followed by BN and PReLU on top of the ResNet50 pre-trained layer, a DDCM module with rates 1, 2, 3 and 4, a bilinear up-sampling of factor 4, an other DDCM module of rate 1, then a bilinear up-sampling, an  $1 \times 1$  convolution and a final bilinear up-sampling.

is the case of the LiDAR acquisition used for this study [15]. Nevertheless, this set is large enough to test our L-DDCM-Net architecture. It covers the Fossard mountain area in upper Mosel valley in Eastern France, where four sectors of 4 km<sup>2</sup> each have been selected. Two of them include arranged areas for walking with a large variety of tracks (*Saint-Mont* and *Cuveaux*). The other ones (*Gris-Mouton* and *Grand-Rupt*) are mostly wood exploitation sectors, the last one featuring some large logging roads. Each DTM image resolution is 4000×4000 pixels (0.5 m pixel size). From these four sectors, two ground truth datasets associated to DTM images are considered:

- Set 1 is a **manual annotation** of each road centerline on the DTM. This set serves as a basis to compare with AMREL results.
- Set 2 is a **semi-automatic segmentation** using AMREL and ILSD softwares. This set is used to train our network.

About the set 1, for each sector, most salient roads were carefully delineated in DTM views. Only the centerline was manually extracted. The achieved polylines constitute the road ground truth [8]. Due to the difficulty to characterize forest roads and to the long and tedious task of manual annotation, it is important to note that this ground truth is not perfect: several road portions have probably been omitted or wrongly annotated. Besides, the representation with centerlines does not provide us the width of a road.

For the set 2, we use the existing tools ILSD and AMREL to generate the forest road ground truth semi-automatically. It allows us to create the data in a simpler, faster and more efficient way. To that end, AMREL is first used to automatically extract road sections. The obtained map is then cleaned to remove obviously bad detections using some image processing tools. Finally, the ground truth map is completed in supervised mode using ILSD software. In particular, we can obtain the road width thanks to the detector of AMREL and ILSD. Contrarily to the set 1, the segmented roads in this dataset are *thick objects*. This enables the network to learn more discriminant features and receptive fields about the detected objects comparing to the centerline representation. Examples of data from both datasets are given in *Fig. 6*.

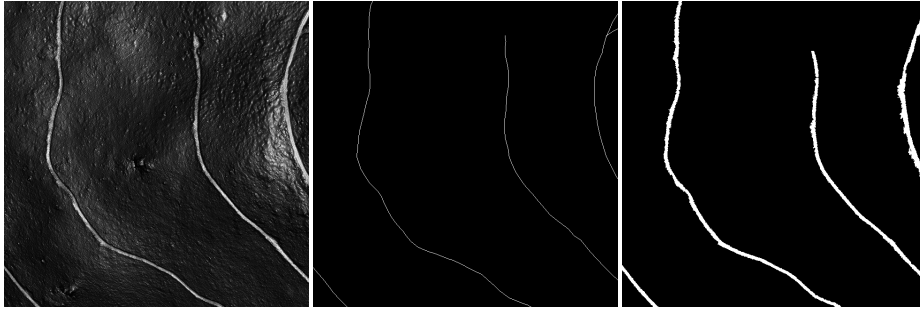


Fig. 6: Sample from training data. Left: shaded DTM image. Middle: Ground truth from set 1 of manual segmentation with centerlines of roads. Right: Ground truth from set 2 of semi-automatic segmentation of forest roads using AMREL and ILSD.

### 3.2 Network training

We implemented the L-DDCM-Net described in *Sec. 2.3* in Python using Tensorflow [23] and Keras [14] libraries which allow to parallelize and optimize many operations via the use of GPU.

For each sector, the DTM image and its corresponding ground truth are split into patches of  $256 \times 256$  pixels without overlap in order to avoid correlation in both training and validation data during the training process. After the splitting process, we obtain 1024 patches in total, about 256 for each sector. Then, data augmentation operations are applied to the resulting patches to enlarge further the data: random rotation with angle in range  $0^\circ$  to  $360^\circ$ , horizontal flip with probability 0.5. Note that this data augmentation is applied only on the training images and performed *on the fly*, *i.e.*, during the training process.

We perform a  $k$ -fold cross-validation, with  $k = 4$ , to estimate the performance of the L-DDCM-Net on the available data. It consists in training  $k$  models by varying the data used for training and testing in such way that the model is tested once on  $k$  different folds. In other words, a sector is evaluated on a model trained with the other three sectors. Four models are trained on groups of patches with a ratio of 80% to the training set and 20% to the validation set.

The whole training process was done on GPU (NVIDIA GeForce RTX 3060 Laptop GPU graphics card with 16Go RAM). We used the Adam optimization algorithm with weight decay (AdamW) [18] with the two parameters  $\beta_1 = 0.9$  and  $\beta_2 = 0.990$  (default values in Tensorflow), and set the weight decay at 0.00005 and the learning rate at 0.00012.

We trained our network for 80 epochs, each epoch being comprised of 76 steps with 8 images per batch. Several values have been tested, and we come out with those values for the smallest BCE loss function on the validation. It should be mentioned that our training is quite fast. It takes about 15 minutes the whole process, and the inference time is about 5 s for each  $4 \text{ km}^2$ -wide sector.

From the predicted probability maps, a threshold of 0.5 is applied to retrieve a binary image in which the white pixels correspond to forest roads and black otherwise. This hyper-parameter can be let to user control according to task

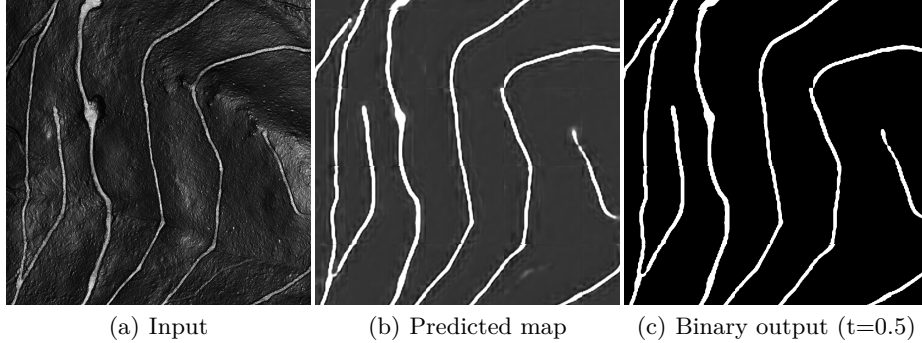


Fig. 7: Example of output of the trained L-DDCM-Net on a part of Gris-Mouton sector.

requirements, as it sets a balance between detecting as many roads as possible and reducing the rate of false detections. We can see in *Fig. 7* an example of output of the trained network.

## 4 Results and discussions

This section shows the experimental results of L-DCMM-Net for forest road segmentation. Experiments are divided into two parts. Firstly, we compare the performance of L-DCMM-Net with other CNN architectures: U-Net [20] and convolutional block attention module (CBAM) [25] on the data of set 2. Secondly, we evaluate the improving performance of L-DDCM-Net to forest road extraction in comparison with the previous work of AMREL on the manual data of set 1.

U-Net [20] is a fully convolutional network. It is originally designed for bio-medical image segmentation, and widely used in other fields. This architecture is well-known for its performance when trained with very few images which is the case of our forest road extraction.

Recently, in many CNN architectures, attention mechanism is usually added in order to make CNNs learn and focus more on important information, rather than learning non-useful background information. Among the different attention modules, convolutional block attention module (CBAM) [25] is well-known for its performance, light-weight and general module. It can be integrated into any CNN architecture seamlessly with negligible overhead and is end-to-end trainable along with the CNN. In order to check whether CBAM modules could help to improve the segmentation results of forest road extraction, they are combined with L-DDCM-Net in two different ways.

- L-DDCM+3-CBAM: a CBAM module is placed at the convolution output of the ResNet block and of both DDCM blocks,
- L-DDCM+8-CBAM: five other CBAM modules are added at the level of each DC module (four for the first DDCM block, one for the second block).

Sector	Method	Time (s)	R (%)	P (%)	F (%)
Saint-Mont	U-Net	7.28	79.40	81.56	80.47
	L-DDCM-Net	<b>5.13</b>	78.47	82.04	80.21
	L-DDCM-Net + 3-CBAM	5.52	<b>80.98</b>	79.99	<b>80.48</b>
	L-DDCM-Net + 8-CBAM	5.89	75.87	<b>82.97</b>	79.26
Gris-Mouton	U-Net	7.49	76.82	<b>98.33</b>	86.25
	L-DDCM-Net	<b>4.86</b>	<b>79.74</b>	95.37	<b>86.85</b>
	L-DDCM-Net + 3-CBAM	5.29	77.19	97.69	86.24
	L-DDCM-Net + 8-CBAM	5.79	77.18	97.55	86.18
Grand-Rupt	U-Net	7.30	76.32	87.89	81.70
	L-DDCM-Net	<b>4.83</b>	<b>78.87</b>	88.77	<b>83.53</b>
	L-DDCM-Net + 3-CBAM	5.26	73.61	91.56	81.61
	L-DDCM-Net + 8-CBAM	5.71	72.05	<b>92.20</b>	80.89
Cuveaux	U-Net	7.50	84.58	86.50	85.53
	L-DDCM-Net	<b>5.08</b>	83.73	89.00	86.28
	L-DDCM-Net + 3-CBAM	5.45	82.91	<b>89.39</b>	86.03
	L-DDCM-Net + 8-CBAM	5.65	<b>86.01</b>	87.55	<b>86.77</b>

Table 1: Comparison of different CNN architectures on data of set 2.

To evaluate the different architectures, we consider the same evaluation metrics as in [8]: recall  $R$ , precision  $P$ , F-measure  $F$  as the harmonic mean of  $R$  and  $P$ . These metrics are given in Eq. 1.

$$R = \frac{\overline{D \cap G_L}}{\overline{G_L}}, \quad P = \frac{\overline{D \cap G_W}}{\overline{D}}, \quad F = \frac{2 * R * P}{R + P} \quad (1)$$

with  $D$  the set of pixels predicted as forest roads,  $G_W$  the set of pixels corresponding to a 20 pixels dilation of ground truth data in set 2,  $G_L$  the set of pixels corresponding to the centerlines of  $G_W$  obtained by Zhang-Suen skeletonization algorithm [27]. The dilated set  $G_W$  is assumed to enclose the real road set and to take into account possible inaccuracy in the detection method.

Tab. 1 summarizes the performance measures of the different architectures on the four test sectors for cross validation. Overall the experiments, L-DDCM-Net gives the fastest inference time. On average, it is about 33% faster than U-Net. This is consistent to the lower complexity of L-DDCM-Net ( $10.4 \times 10^6$  trainable parameters) compared to U-Net ( $18.9 \times 10^6$ ). Adding CBAM blocks to L-DDCM-Net slightly increases the inference time and the model complexity ( $12.5 \times 10^6$  trainable parameters). Note that the training time is divided by 3 with L-DDCM-Net ( $\simeq 15$  minutes) compared to U-Net or L-DDCM-Net with CBAM blocks ( $\simeq 45$  minutes).

Regarding the evaluation measures, U-Net and L-DDCM-Net show similar results, with small variations depending on the test area. There is not a clear

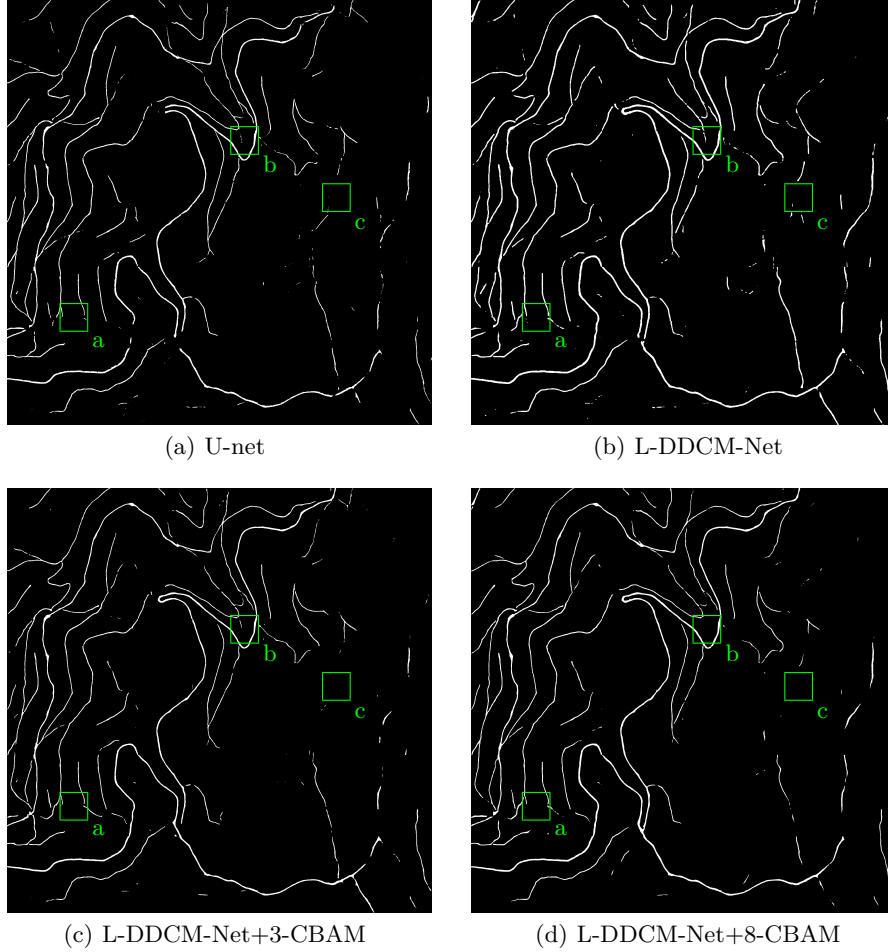


Fig. 8: Binary predicted maps obtained by different network architectures on Gris-Mouton sector. The networks are trained and tested with data from set 2. Hard configuration examples of Fig. 9 are located in green.

difference when adding CBAM blocks to L-DDCM-Net. On wood exploitation sectors, Gris-Mouton and Grand-Rupt, achieved higher precision is balanced with recall decrease. But this observation does not hold anymore on the two other sectors, where both tested architectures for CBAM show contradictory results. However, better tuning for hyper-parameters could possibly be found to draw more benefits from attention modules for this task of forest road detection.

Fig. 8 gives an example of prediction maps. We can observe that these maps, produced by the trained models, give a plausible forest road network. Some false positives match well visible road-like objects, that were not selected as ground truth because considered as short isolated sections or impracticable roads. Many

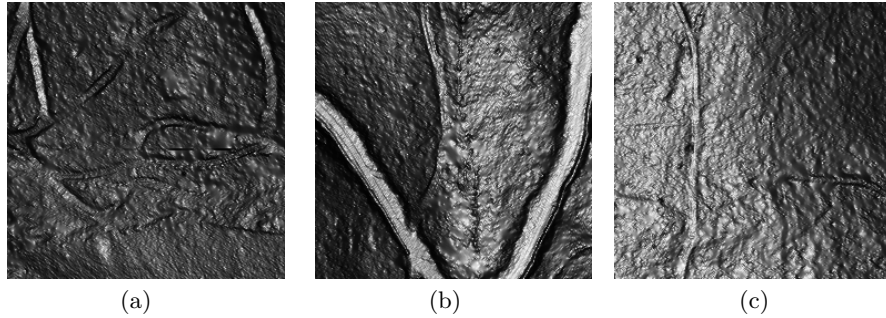


Fig. 9: Hard configurations for the detection task: a) DTM interpolations due to point lack, b) steep section at road junction, c) undisclosed road in flat area.

Sector	Method	R (%)	P (%)	F (%)
Saint-Mont	AMREL	67.46	74.90	70.99
	L-DDCM-Net	<b>76.72</b>	<b>79.01</b>	<b>77.85</b>
Gris-Mouton	AMREL	69.87	88.13	77.94
	L-DDCM-Net	<b>77.46</b>	<b>91.74</b>	<b>84.00</b>
Grand-Rupt	AMREL	73.01	78.37	75.60
	L-DDCM-Net	<b>78.44</b>	<b>86.94</b>	<b>82.47</b>
Cuveaux	AMREL	68.73	80.49	74.14
	L-DDCM-Net	<b>80.61</b>	<b>87.32</b>	<b>83.83</b>

Table 2: Comparison between AMREL [8] and L-DDCM-Net using data in set 1 of manual road annotation with centerlines for the evaluation.

undetected parts are connected to blurred areas in the DTM image possibly due to a lack of ground points (see *Fig. 9 a*). Others correspond to narrow steep or damaged roads left back to erosion or vegetation (see *Fig. 9 b*). Moreover, roads crossing flat areas at almost the same height are undisclosed in this low contrasted part of the slope-shaded map (see *Fig. 9 c*). Already hard to delimit during manual digitization, such objects have few chance to be retrieved successfully by the network.

It must be noted that the measured performance values should be taken with caution. Many processes during the training of a deep network model are based on random selections, so that this step is certainly not reproducible. For each tested configuration, only the best model obtained on a small series of trials was kept. Therefore, we mostly conclude that among the 4 network architectures evaluated, L-DDCM-Net provides faster inference speed while maintaining high accuracy in detection. Hereafter, we use L-DDCM-Net as reference model for the next experiments.

To evaluate the improvement and the efficiency of the L-DDCM-Net for forest road extraction, we compare the results of the trained models with those of

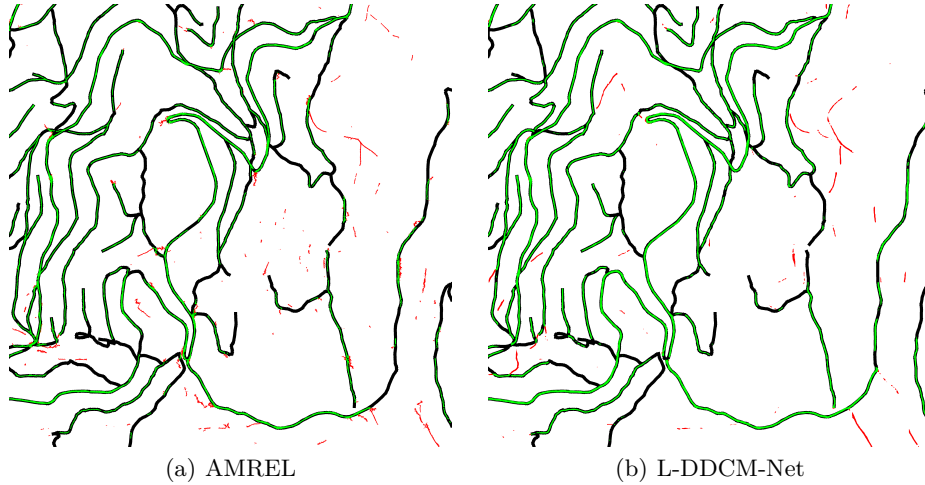


Fig. 10: Comparison between AMREL (a) and L-DDCM-Net (b) using the ground truth data from set 1. Precision maps: dilated ground truth in black, good detection in green, false detection in red.

AMREL on the data in set 1. For this, we consider the same evaluation metrics as in previous experiments (see *Eq. 1*). In particular, the set  $G_L$  is the centerlines of data in set 1 and  $G_W$  corresponds to a 28 pixels dilation of  $G_L$ . The comparison of both methods is reported in *Tab. 2*. L-DDCM-Net outperforms AMREL. L-DDCM-Net presents, on average, a performance increase of 8.5% in recall and 5.8% in precision on the four sectors. It takes 5 s to process a sector using L-DDCM-Net. Although it is not possible to compare this execution time with the 30 s reported for AMREL in [7] run on slower hardware, up to now, it is not guaranteed that this last could be adapted to GPU programming.

*Fig. 10* illustrates the results on one of the sectors. We observe less discontinuities and false detection in L-DDCM-Net results than in AMREL. This makes the results more favorable and complete in this task of forest road detection.

All implementation of L-DDCM-Net and execution procedure to reproduce the test results of Gris-Mouton sector in *Tab. 1* and *2* are available at:

<https://github.com/paulgeorges1998/Light-DDCM-Net>.

Because other sectors cross well-known archaeological areas, it is unfortunately not possible to let a full access to the whole data set. For the test of the three other sectors and the training on all sectors, access to relevant DTM tiles may be requested to the LiDAR data owner.

## 5 Conclusion

This paper addresses the forest road detection in LiDAR data using deep learning approaches. More precisely, we consider digital terrain model (DTM) images computed from an interpolated surface of the 3D LiDAR point cloud. Two



datasets, one from manual annotation with centerline and the other from a semi-automatic extraction using two softwares AMREL and IDSL, are used for training, validating and testing the method. Different convolution neural networks (CNNs) are considered in this work: L-DDCM-Net, U-Net and CBAM attention blocks. The trained network models are also compared with the AMREL results to show the improvement and the efficiency of the deep learning approaches for the problem of forest road detection.

In general, the CNNs provide high detection accuracy in addition to a fast prediction time. However, they also have several limitations in the present case. Firstly, we do not observe significant improvements by modifying the different architectures: adding attention modules, modifying the hyper-parameters, . . . This may be due to the restricted amount of data used for training the CNN. In order to improve the CNN performance, we may need to provide more meaningful information to the network. Observing the false negative results, undetected forest roads often correspond to narrow steep sections, that are less contrasted in slope images. For traffic convenience, most of forest roads do not deviate significantly from isolines. In order to guide the networks to learn such behavior, we suggest providing not only the slope intensity, but also its direction as additional training data. The slope direction can be computed from the triangulated mesh used to build the DTM, and represented as a second input image. Furthermore, bigger size of training images could be considered to better capture the global context of the forest roads. Next availability of more general purpose LiDAR data at country-wide scale [13] will hopefully facilitate a complete reproducibility of this kind of work, including training stages. Combined with digitization facilities brought by AMREL and ILSD softwares, it will contribute to the production of such a wider training dataset.

**Acknowledgements:** DTM images are derived from Fossard LiDAR data acquired in scope of the PCR AGER project (*Projet collectif de recherche "Archéologie et Géoarchéologie du premier Remiremont et de ses abords"*), dir. Charles Kraemer, HISCANT Laboratory, Université de Lorraine.

## References

1. Clode, S., Rottensteiner, F., Kootsookos, P., Zelnicker, E.: Detection and vectorization of roads from lidar data. *Photogrammetric Engineering & Remote Sensing* **73**(5), 517–536 (2007). <https://doi.org/10.14358/PERS.73.5.517>
2. David, N., Mallet, C., Pons, T., Chauve, A., Bretar, F.: Pathway detection and geometrical description from ALS data in forested montaneous areas. *International Archives of Photogrammetry, Remote Sensing and Spatial Information Sciences* **38**(part 3/W8), 242–247 (2009)
3. Debled-Rennesson, I., Feschet, F., Rouyer-Degli, J.: Optimal blurred segments decomposition of noisy shapes in linear time. *Computers and Graphics* **30**(1), 30–36 (2006). <https://doi.org/10.1016/j.cag.2005.10.007>
4. Even, P., Grzesznik, A., Gebhardt, A., Chenal, T., Even, P., Ngo, P.: Interactive extraction of linear structures from LiDAR raw data for archaeomorphological

- structure prospection. The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences **XLIII-B2-2021**, 153–161 (2021). <https://doi.org/10.5194/isprs-archives-XLIII-B2-2021-153-2021>
5. Even, P., Ngo, P.: AMREL: Automatic mountain road extraction from LiDAR data, <https://github.com/evenp/AMREL.git/>, 2022
  6. Even, P., Ngo, P.: ILSD: Interactive linear structure detector, <https://github.com/evenp/ILSD.git/>, 2022
  7. Even, P., Ngo, P.: Live extraction of curvilinear structures from LiDAR raw data. ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences pp. 211–219 (2020). <https://doi.org/10.5194/isprs-annals-V-2-2020-211-2020>
  8. Even, P., Ngo, P.: Automatic forest road extraction from lidar data of mountainous areas. In: First Joint Conference on Discrete Geometry and Mathematical Morphology. pp. 93–106 (2021). [https://doi.org/10.1007/978-3-030-76657-3\\_6](https://doi.org/10.1007/978-3-030-76657-3_6)
  9. Even, P., Ngo, P., Kerautret, B.: Thick line segment detection with fast directional tracking. In: Proc. of 20th International Conference on Image Analysis and Processing. pp. 159–170 (2019). [https://doi.org/10.1007/978-3-030-30645-8\\_15](https://doi.org/10.1007/978-3-030-30645-8_15)
  10. Ferraz, A., Mallet, C., Chehata, N.: Large-scale road detection in forested mountainous areas using airborne topographic lidar data. ISPRS Journal of Photogrammetry and Remote Sensing **112**, 23–36 (2016). <https://doi.org/10.1016/j.isprsjprs.2015.12.002>
  11. Goodfellow, I., Bengio, Y., Courville, A.: Deep Learning. MIT Press (2016)
  12. Hui, Z., Hu, Y., Jin, S., Yevenyo, Y.Z.: Road centerline extraction from airborne LiDAR point cloud based on hierarchical fusion and optimization. ISPRS Journal of Photogrammetry and Remote Sensing **118**, 22–36 (2016). <https://doi.org/10.1016/j.isprsjprs.2016.04.003>
  13. IGN: LIDAR HD – Une cartographie 3D du sol et du sursol de la France, <https://geoservices.ign.fr/lidarhd/>, last check : December 2022.
  14. Keras: Python interface for artificial neural networks, <https://keras.io/>, 2022
  15. Kraemer, C., Cartier, D., Chenal, T., Delangle, C., Gebhardt, A., Grzesznik, A., Harmand, D., Hingray, T., Poszwa, A.: ArchéoGéographie du premier Remiremont et de ses abords : le Saint-Mont et le massif du Fossard. Rapport de Projet AGER 1, Université de Lorraine (2018)
  16. Liu, Q., Kampffmeyer, M., Jenssen, R., Salberg, A.B.: Road mapping in LiDAR images using a joint-task dense dilated convolutions merging network. In: IEEE International Geoscience and Remote Sensing Symposium. pp. 5041–5044 (2019). <https://doi.org/10.1109/IGARSS.2019.8900082>
  17. Liu, Q., Kampffmeyer, M., Jenssen, R., Salberg, A.B.: Dense dilated convolutions’ merging network for land cover classification. IEEE Transactions on Geoscience and Remote Sensing **58**(9), 6309–6320 (2020). <https://doi.org/10.1109/TGRS.2020.2976658>
  18. Loshchilov, I., Hutter, F.: Decoupled weight decay regularization. In: 7th International Conference on Learning Representations (2019). <https://doi.org/10.48550/arXiv.1711.05101>
  19. Merveille, O., Naegel, B., Talbot, H., Najman, L., Passat, N.: 2D filtering of curvilinear structures by ranking the orientation responses of path operators (RORPO). Image Processing On Line **7**, 246–261 (2017). <https://doi.org/10.5201/ipol.2017.207>
  20. Ronneberger, O., Fischer, P., Brox, T.: U-net: Convolutional networks for biomedical image segmentation. In: 18<sup>th</sup> International Conference on Medical Im-

- age Computing and Computer-Assisted Interventions. pp. 234–241 (10 2015). [https://doi.org/10.1007/978-3-319-24574-4\\_28](https://doi.org/10.1007/978-3-319-24574-4_28)
21. Russakovsky, O., Deng, J., Su, H., Krause, J., Satheesh, S., Ma, S., Huang, Z., Karpathy, A., Khosla, A., Bernstein, M., Berg, A.C., Fei-Fei, L.: ImageNet large scale visual recognition challenge. *International Journal of Computer Vision (IJCV)* **115**(3), 211–252 (2015). <https://doi.org/10.1007/s11263-015-0816-y>
  22. Salberg, A.B., Trier, Ø.D., Kampffmeyer, M.: Large-scale mapping of small roads in lidar images using deep convolutional neural networks. In: *Scandinavian Conference on Image Analysis*. pp. 193–204 (2017). [https://doi.org/10.1007/978-3-319-59129-2\\_17](https://doi.org/10.1007/978-3-319-59129-2_17)
  23. TensorFlow: Software library for machine learning and artificial intelligence, <https://www.tensorflow.org/>, 2022
  24. White, R.A., Dietterick, B.C., Mastin, T., Strohman, R.: Forest roads mapped using LiDAR in steep forested terrain. *Remote Sensing* **2**(4), 1120–1141 (2010). <https://doi.org/10.3390/rs2041120>
  25. Woo, S., Park, J., Lee, J.Y., Kweon, I.S.: CBAM: Convolutional block attention module. In: *Proceedings of the European Conference on Computer Vision*. pp. 3–19 (2018). [https://doi.org/10.1007/978-3-030-01234-2\\_1](https://doi.org/10.1007/978-3-030-01234-2_1)
  26. Yu, F., Koltun, V.: Multi-scale context aggregation by dilated convolutions. In: *International Conference on Learning Representations* (2016). <https://doi.org/10.48550/arXiv.1511.07122>
  27. Zhang, T.Y., Suen, C.Y.: A fast parallel algorithm for thinning digital patterns. *Communications of the ACM* **27**(3), 236–239 (1984). <https://doi.org/10.1145/357994.358023>