# Road Detection and Segmentation from Aerial Images using a CNN based System

Loretta Ichim
Faculty of Automatic Control and Computers
Politehnica University of Bucharest
"Stefan S. Nicolau" Institute of Virology
Bucharest, Romania
Email: loretta.ichim@upb.ro

Dan Popescu
Faculty of Automatic Control and Computers
Politehnica University of Bucharest
Bucharest, Romania
Email: dan.popescu@upb.ro

***Abstract*—This paper proposes a system architecture based on deep convolutional neural network (CNN) for road detection and segmentation from aerial images. These images are acquired by an unmanned aerial vehicle implemented by the authors. The algorithm for image segmentation has two phases: the learning phase and the operating phase. The input aerial images are decomposed in their color components, preprocessed in Matlab on Hue channel and next partitioned in small boxes of dimension 33 × 33 pixels using a sliding box algorithm. These boxes are considered as inputs into a deep CNN. The CNN was designed using MatConvNet and has the following structure: four convolutional layers, four pooling layers, one ReLu layer, one full connected layer, and a Softmax layer. The whole network was trained using a number of 2,000 boxes. The CNN was implemented using programming in MATLAB on GPU and the results are promising. The proposed system has the advantage of processing speed and simplicity.**

***Keywords*—aerial images; convolutional neural networks; road detection; segmentation; unmanned aerial vehicles (UAV)**

## I. INTRODUCTION

The extraction of reliable information from aerial images is a difficult problem, but it has numerous important utilizations: the disaster monitoring (earthquakes, floods, vegetation fires, etc.), crop monitoring in precision agriculture, border surveillance, traffic monitoring, and so on. In aerial monitoring of ground surfaces, the detection and segmentation of roads represent an important challenge. To this end, different image processing techniques were considered. Texture analysis techniques are used to detect and segment regions of interest and, particularly roads, from aerial images in [1-3] but the choice of the representative features depends on the specific context of the application that uses it. The authors in [4] consider also a supervised learning approach to detect road textures using a neural network.

To detect and segment the roads, concatenated images, created by photomosaic generation, can be useful. Thus, the gaps or duplications of regions, as they may appear in the collection of images taken, are avoided. In this case, the UAV is cheaper and more flexible solution (because it ensures superior image resolution even under adverse weather conditions). Recently, real time image processing in videos taken from low-/mid-altitude UAV (multi-copter type) is proposed in [5] for efficient road detection and tracking. The authors used as methodology design the Gaussian Mixture Model, structure tensor, and GraphCut. Different road features and information as the Stroke Width Transform, colors, and width, are combined to highlight possible road candidates [6]. Then a Gaussian Mixture Model is built to classify these candidates as road and background. Starting from these road and background classes, Convex Active Contour model segmentation is proposed to extract whole road regions. In order to increase the accuracy and robustness of road detection in [7] a deep Convolutional Neural Network (CNN) was successfully used. For efficient training, in this project the authors proposed the parallel image processing in GPU. They test different nets which were trained using DIGITS (a training system Web App) to determine the best architecture. Also, a road structure refined CNN (RSRCNN) approach for automatic road extraction in aerial images was proposed in [8]. Recently, in [9] the authors developed a semantic segmentation of buildings and roads from aerial images based on CNN architecture.

In this paper we proposed a system able to segment the roads from aerial images taken with a fixed wing UAV. The system is based on a CNN architecture [10] using a supervised learning algorithm. CNN was designed using MatConvNet [11]. Compared to the algorithms presented in literature, ours has the advantage of simplicity and accuracy.

## II. METHODOLOGY

### A. System Architecture

The system for the road detection and segmentation from aerial images is presented in Fig. 1 and contains two main modules: UAV module (fixed wing type) and GROUND module. The images taken from UAV' camera, are transmitted via digital data link to GROUND module. In order to detect and segment the roads, successive images are taken with constant rate on the programmed trajectory. The images are saved in the Image Buffer in order to be next processed. The image is firstly decomposed in color components and, for shadow attenuation, only H (Hue) component is considered. Two important operations of primary image processing are included: noise rejection and contrast enhancement. Therefore, the median filter (kernel 3 × 3), CLAHE (contrast - limited

adaptive histogram equalization), and a transformation for sharpening the edges were used. For CLAHE [12], the contrast enhancement limit was set to 0.5 and the distribution was set to 'Rayleigh'. After primary image processing the sliding box decomposition of the image is made. The box size is of 33 × 33 pixels with a sliding step of 1 pixel. Thus, the input data for the proposed convolutional neural network are monochrome images (corresponding to H component) of 33 × 33 pixels. These are passed through the entire network to get their classification into two classes: ROAD and NON-ROAD.

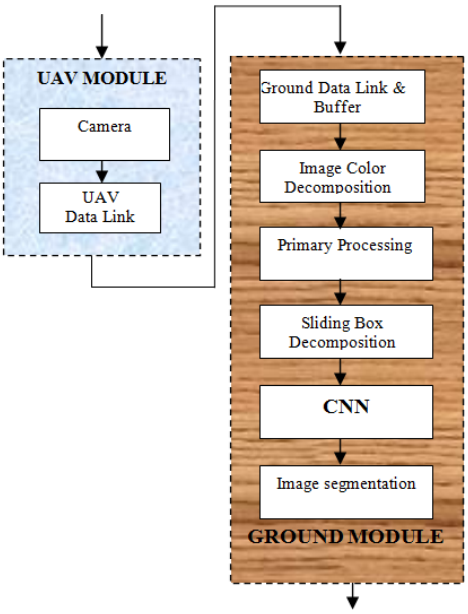


Fig. 1. The system for the road detection and segmentation (block diagram).

## B. CNN Architecture

The proposed architecture of CNN contains four convolutional layers followed by four layers of spatial reduction (pooling layers), one fully connected layer, one ReLu activation layer and a Softmax layer (Fig. 2). It is a custom created neural network structure, which was trained in a supervised mode. The input images were selected from the UAS implemented by the authors in the research project MUROS [13]. The result of a CNN-specific operation (convolution or pooling) is defined as a map of features. Each of the results obtained will have a third dimension, the depth (the number of neurons) of each layer. The network functionality is based on two phases: the learning phase and the operating phase. All the primary processing steps applied on the training set will be performed on the testing set too. In order to provide a large dataset, boxes of 33 × 33 pixels were extracted in the learning phase, from each processed image. The box dimension is experimentally chosen taken into account the image resolution, the width, and the nature of the road. Note that larger boxes can be used for higher resolution images or larger road. Two labels, which represent the network output (classes), are used for the boxes: ROAD - positive and NON-ROAD – negative (Fig. 4). Each box is labeled using the manual segmentation provided. A positive box has the central pixel as ROAD. Each positive box has four associated learning boxes: the initial box and boxes rotated with 90, 180, and 270 degrees (Fig. 4.a). In Fig. 4.b are also presented two examples of negative samples. Each box is labeled using the manual segmentation provided.

The CNN receives preprocessed images on H component (boxes of dimension 33 × 33 pixels) from the Sliding Box Decomposition module. The input is not considered as a network layer. The first layer is a convolutional layer of 3 × 3 pixels, with the stride 1 and the padding equal to [1010]. This means that the results are boxes with dimension of 32 × 32 pixels. Thus, next successive divisions with 2 and pooling operations without losses are permitted. The layer contains 20 filters considered as neurons. They are initialized with random numbers from a Gaussian distribution. The second layer in the CNN structure is a pooling layer which reduces the space dimension by half with a sliding box of 2 × 2 pixels. The step is also 2 pixels because there is no need to overlap two blocks. We used pooling layers that reduce the size by keeping the pixel average of the sliding box. The second convolutional layer does not change the size of the feature maps, but through it the deep is increased to 50 neurons. Its parameters are similar to those of the first convolutional layer, with the difference that the classical padding (P = 1) is made in order to preserve the box size. The corresponding pooling layer is the same as the precedent, reducing to half the size of the feature mapping from the previous layer. Using similar structures, after a ReLu layer (rectified linear unit), boxes of dimension 3 × 3 and deep of 100 filters are obtained. The last convolutional layer is a fully connected layer which reduces the spatial resolution to a spatial resolution of 1 × 1 and a depth of 2 (corresponding to labels ROAD and NON-ROAD). The last layer of this neural network architecture is a layer called Softmax. This is a "loss" layer because a loss function measures how badly the network is doing for the input.

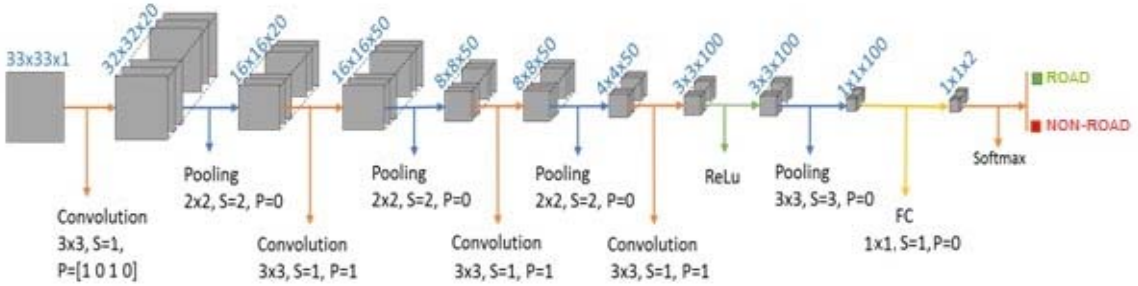


Fig. 2. The system for the road detection and segmentation (block diagram).

In the classification process (operating phase) the CNN receives an input box, with the same size like the training box (33 × 33 pixels), and returns, for the central pixel, a score for each class. For each pixel, we consider the class with the higher score. The pixels classified by CNN are lastly considered for the segmented image reconstruction (Image segmentation module - Fig. 1). In this module the pixels are integrated in a novel image (the segmented image). Now, morphological operations (errosion and dilation) are made in order to eliminate possible noises due to segmentation process.

## III. Experimental Results

The images were captured along a path generated from parallel lines, with distances between lines of 75 m, altitude of 200 m, and speed of 70 km/h. Camera type was Sony Nex7, objective 50 mm, 24.3 megapixels, and 10 fps. The images were taken by a fixed wing type UAV (MUROS – Fig. 3.a) in a real flight. The camera was mounted in a gyro-stabilized payload (Fig. 3.b). The images were generated with Agisoft Photoscan Professional Edition [14]. The primary processing filters and color component have very efficient effects on the image. Therefore, the primary processing of images was used both in the learning phase and in the operating phase. The CNN was also implemented in the Ground module, on a computing system for target image evaluation which has the following characteristics: Intel Core i7-4790 CPU, 4.00 GHz, 16.0 GB RAM, Windows 8.1, x 64, GPU Programming in MATLAB [15]. For CNN we used GPU implementation which ensures high processing speed. The segmentation time was about of 10 s for an image.

The training files are represented in MatConvNet [11] by .mat files, each file containing a number of lines. Each line is composed by concatenating the lines from a box and also contains the box label. Thus, a training file is a structure with two fields: data and labels. Each line from a training file contains a vector of 1089 elements, in the data field, which represents a positive or a negative box (33 × 33 pixels), and also the corresponding label, in the label field. For training we used 1000 boxes with asphaltic road (examples in Fig. 4.a) and 1000 boxes with non-road (trees, grass, crops and ground - examples in Fig. 4.b) from 50 images. We trained the proposed CNN for 50 epochs (experimentally chosen). For a greater number of epochs, it can be seen (Fig. 5) that both the objective function and the errors decrease very little, while the learning time increases. We can consider that the network mostly completed their training within 50 epochs. The learning rate varied between 0.0005 and 0.005 depending of epoch number. Note that the Fig. 5 represents in the left graphic the training error (blue) and the validation error (green) across the epochs. The kernel sizes and the number of filters per layer (the deep) were also experimentally and lead to the best road segmentation accuracy. This CNN is generally robust to changes but it performs better in zones without trees (covering the road) and buildings (concrete constructions or roofs).

Fig. 6 presents the results obtained for two images from a testing set. The images from the left are the original images: upper it is a narrow asphalt road and bottom is a highway intersected by narrow asphalt road. The images from right represent the segmented images of the previous images (road with white and the rest with black). The CNN output gives pixel by pixel information on what category each fall into (ROAD or NON-ROAD). Road pixels are labeled with ”1s” (white) and non road pixels are labeled with ”0s” (black). Note that, in this case, we trained CNN with asphalt type road. For other type of roads, a new training is necessary. Most of the noise created by false-positives and false-negatives is rejected during post-processing step in the “Image segmentation” module of the system (Fig. 1).

a) b)

Fig. 3. Image acquisition system: a) MUROS UAV for real - life image acquisition, b) payload with camera.

a) b)

Fig. 4. Examples of boxes used for training (positive boxes, asphaltic road – Fig. 4.a and negative boxes – Fig. 4.b).

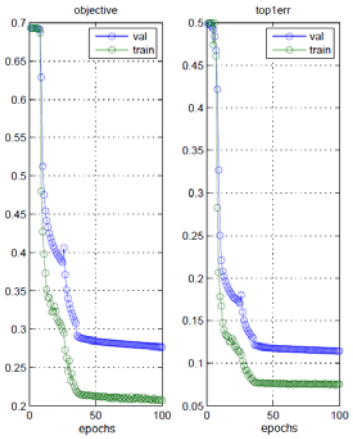


Fig. 5. Objective function and error versus epochs.

Taken into account the performance statistics [16], the precision - *PPV* (Positive Predictive Value) - (1) and the accuracy - *ACC* - (2) were calculated from a set of 21901 sample boxes from the images Im 01 and Im 02 (Fig. 6), where *TP* is true positive, *TN* is true negative, *FP* is false positive, and *FN* is false negative. For the classified pixels of these images, the system performances are presented in Table I, and an average ACC is of 98.8%.

$$PPV = \frac{TP}{TP + FP} \quad (1)$$

$$ACC = \frac{TP + TN}{TP + TN + FP + FN} \quad (2)$$

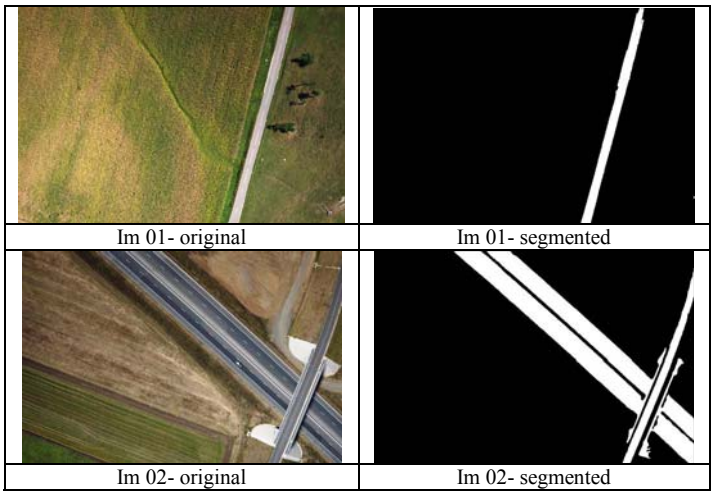


Fig. 6. Output for asphaltic road segmentation.

The performances of road segmentation depend on the altitude of flight (low or mid-altitude), image resolution, and also on the CNN structure.

The comparison with other similar works are presented in Table II and indicates good results for the proposed method. As it was specified earlier, in [7] an improved SegNet CNN architecture, in [5] a static GraphCut method, and in [6] a Convex Active Contour model are proposed for road segmentation.

TABLE I. PERFORMANCE METRICS OF THE PROPOSED SYSTEM

| Indicator/ Image | TP [×10³] | TN [×10³] | FP [×10³] | FN [×10³] | PPV [%] | ACC [%] |
|---|---|---|---|---|---|---|
| 01 | 681 | 23307 | 1 | 11 | 99.8 | 99.9 |
| 02 | 3862 | 19996 | 139 | 3 | 96.5 | 99.4 |
| Average 100 images | - | - | - | - | 97.2 | 98.8 |

TABLE II. COMPARISON WITH PERFORMANCE METRICS OF OTHER WORKS

| Indicator/ Method | Resolution [pixels] | Time [s] | PPV [%] | ACC [%] |
|---|---|---|---|---|
| [7] | 16×375×375 | - | - | 72 |
| [5] | 1046×595 | 0.029 | 98.41 | - |
| [6] | 617×411 | 0.857 | 91.5 | - |
| Proposed method | 6000×4000 | 1.512 | 97.2 | 98.8 |

## IV. CONCLUSIONS

Road detection is a difficult task in aerial image segmentation due to different size and texture. One of the most important steps in training a CNN is the preprocessing stage. In the case of road segmentation, noise rejection and contrast enhancement techniques had been applied. The second important stage is the selection of the training data. The selected boxes have to cover all the road types from the over-flown area (thin and thick roads, with ramifications or without ramifications). Augmentations aren't needed in this case because there are enough training samples (iterating over the image with a sliding 33 × 33 window with stride 1, will generate enough training samples). The proposed system for road detection and segmentation has the advantage of processing speed, simplicity and possible application to pipeline or river segmentation from aerial images.

## ACKNOWLEDGMENT

This work has been funded by UEFISCDI, Bridge Grant Program, project SIMUL, BG49/2016 and Romanian National Authority for Scientific Research and Innovation.

## REFERENCES

[1] D. Popescu and L. Ichim, "Aerial image segmentation by use of textural features," In Proc. 20th International Conference on System Theory, Control and Computing (ICSTCC), Sinaia, Romania, pp. 721–726, October 2016.

[2] H. Kong, J.Y. Audibert, and J. Ponce, "General road detection from a single image," in IEEE Trans. Image Process. vol. 19, pp. 2211–2220, 2010.

[3] D. Popescu and L. Ichim, "Image recognition in UAV application based on texture analysis," in ACIVS 2015. LNCS, Springer, Heidelberg, vol. 9386, pp. 693–704, 2015.

[4] V. Mnih and G.E. Hinton, "Learning to detect roads in high-resolution aerial images," In ECCV 2010. LNCS, Springer, Heidelberg, vol. 6316, pp. 210–223, 2010.

[5] H. Zhou, H. Kong, L. Wei, D. Creighton, and S. Nahavandi, "Efficient road detection and tracking for Unmanned Aerial Vehicle," IEEE Trans. on Intell. Transportation Systems, vol. 16(1), pp. 297–309, Feb. 2015.

[6] H. Zhou, H. Kong, L. Wei, D. Creighton, and S. Nahavandi, "On detecting road regions in a single UAV image," IEEE Trans. on Intell. Transportation Systems, vol. 18(7), pp. 1713–1722, July 2017.

[7] T. Ayoul, T. Buckley, and F. Crevier, "UAV navigation above roads using convolutional neural networks," http://cs231n.stanford.edu/reports/2017/pdfs/553.pdf, 2017.

[8] P. Kaiser, J.D. Wegner, A. Lucchi, M. Jaggi, T. Hofmann, and K. Schindler, "Learning aerial image segmentation from online maps," IEEE Transactions on Geoscience and Remote Sensing, vol. 55(11), pp. 6054–6068, Nov. 2017.

[9] Y. Wei, Z. Wang, and M. Xu, "Road structure refined cnn for road extraction in aerial image," IEEE Geoscience and Remote Sensing Letters, vol. 14(5), pp. 709–713, 2017.

[10] Convolutional neural networks for visual recognition. http://cs231n.github. io/understanding-cnn/.

[11] A. Vedaldi, K. Lenc, and A. Gupta, "MatConvNet convolutional neural networks for MATLAB," https://arxiv.org/pdf/1412.4564.pdf, 2018.

[12] CLAHE, *Contrast Limited Adaptive Histogram Equalization;* Site: https://www. mathworks.com/help/images/ref/adapthisteq.html; Accesat 2017.

[13] MUROS, https://trimis.ec.europa.eu/project/multisensory-robotic-system-aerial-monitoring-critical-infrastructure-systems.

[14] Agisoft PhotoScan, www.agisoft.com.

[15] http://www.mathworks.com/products/matlab/.

[16] M. Sokolova and G. Lapalme, "A systematic analysis of performance measures for classification tasks," Inf. Process. Manage., vol. 45, pp. 427-437, July 2009.