

# Road detection with EOSResUNet and post vectorizing algorithm

Oleksandr Filin  
EOS Data Analytics

alexandr.filin@eosda.com

Anton Zapara  
EOS Data Analytics

anton.zapara@eosda.com

Serhii Panchenko  
EOS Data Analytics

sergey.panchenko@eosda.com

## Abstract

*Object recognition on the satellite images is one of the most relevant and popular topics in the problem of pattern recognition. This was facilitated by many factors, such as a high number of satellites with high-resolution imagery, the significant development of computer vision, especially with a major breakthrough in the field of convolutional neural networks, a wide range of industry verticals for usage and still a quite empty market. Roads are one of the most popular objects for recognition. In this article, we want to present you the combination of work of neural network and postprocessing algorithm, due to which we get not only the coverage mask but also the vectors of all of the individual roads that are present in the image and can be used to address the higher-level tasks in the future. This approach was used to solve the DeepGlobe Road Extraction Challenge.*

## 1. Introduction

Solving the problems of computer vision has made a significant progress due to the development of convolutional neural networks. And it would be absolutely wrong to reject this approach. However, there is the question: Is the result of the neural network sufficient to provide both high-quality and useful data that could be used to solve more complex problems?

It is certain that a well-prepared neural network can give a very qualitative result in the form of a coverage mask, but there are only a few options for further use of such mask, so it's worth thinking about doing something more.

Within the roads recognition problem, we focused on the obtaining of qualitative information and some stats, which can be used for various types of manipulation in the future. For example, such manipulations may include the addition of the resulting road mask or the ability to completely redraw a road mask, knowing some characteristics of each of the roads separately. From the mask that was output from the neural network, we get the resulting information by post-processing.

Before describing our method of post-processing we

need to formulate some rules by which we evaluate what can be called a high-grade road and what characteristics it should have.

Firstly, we need to decide on the very notion of a "road". In our concept, one road is a "straight" vector, which ends either on the connection with another road - the crossroad, or when it forms a sharp turn. "Straight" will be considered any vector that does not change its direction more than  $45^\circ$ . Such change of direction will be considered as a turn of the road.

Secondly, we can absolutely say that the roads are a closed graph. There are no roads that appear out of nowhere and just do not lead to anywhere. We can say that any road must be connected with some other road, or in the case of a limited image - collide with the border. Of course there are some exceptions to this concept (for example roads located between water sources). But in practice, such situations are extremely rare. There were no similar situations have found in the DeepGlobe Road Extraction Challenge dataset.

Thirdly, each road has different attributes, such as length, width, coating and many others. However, in this article we will confine ourselves to using only a few of them. For example, we may need knowledge about the road width, which is often almost unchanged throughout the entire length. We can draw such a conclusion by conducting some analysis.

Using all of this knowledge we can look at the problem as at the source of a number of useful information that we will try to get.

Nevertheless, it is necessary not to forget about the neural network which provides a starting material for further postprocessing. In order to obtain a qualitative result of postprocessing, it is necessary to take care of the competent architecture of the neural network and to focus it on the solution of the above task.

The results of the work of this neural network and post-processing were presented in the DeepGlobe Road Extraction Challenge competition. As a result of the post-processing algorithm, we managed to improve the mask of the final dataset by 0.2%. The gain is not great, but it is easily explained by a very noisy dataset, which will be

discussed in detail later. But most importantly, this algorithm provides a wide range of possibilities for modifying the resulting mask, and a large amount of useful information, such as a list of roads and the characteristics of each of them separately. These results have the applied meaning and can be used for higher-level tasks such as cartography, logistics, etc.

## 2. Related Work

The task of the roads recognition is a task of semantic segmentation. There was a sufficiently large number of experiments made in this direction and everybody attempt to solve this task in completely different ways. The simplest solutions are based on a multilayer perceptron. An example of such solution is the approach described in the article by Kahraman et al. [7]. However, we found the encoder-decoder neural network architecture more preferable. The most popular of these are UNet [10] and SegNet [1]. Networks based on these two architectures quite often won in various competitions related to the processing of satellite images. For example, SpaceNet, including the competition for finding roads, Understanding Amazon from space on Kaggle. Also, the frequent approach is to use pre-trained models as an encoder. The bright representatives of this approach are the networks TernaNet [6] and LinkNet [2]. Another interesting modification of the UNet architecture is Residual Unet [12], which includes Residual blocks [5]. Also worth mentioning two articles based on OpenStreetMap [11]: Generative Street Addresses from Satellite Imagery [3] and Enhancing Road Maps by Parsing Aerial Images Around the World [8] that describe the problem of finding roads, and methods of solving it using OSM as a groundtruth.

## 3. Architecture of Neural Network

The basis of our neural network (Figure 1) was taken from UNet architecture, it includes 5 blocks of encoder and decoder, each of which is a Residual block and passes the input signal straight to the next one. Upsample layers are represented as standard Deconvolutional layers and take a direct signal from Pooling layers of the same level. Another important modification is the optimization of IoU [9] instead of entropy, as it is done in most cases. And to increase the speed of the algorithm is also used a quite interesting trick: the neural network is trained on the 256x256 images but can predict the result for images of any other sizes. This method does not affect the architecture of the neural network, because there is the same amount of information is stored in one pixel of the image as in 1024x1024 images, but we can qualitatively augment our dataset and speed up the learning of the model.

## 4. Training

### 4.1. Dataset overview

The DeepGlobe Road Extraction Challenge [4] dataset is presented in the form of 6226 satellite images 1024x1024 in RGB. Each picture is accompanied by its mask, where the background is marked with a black color, and roads with a white color (Figure 2).

But it's worth noting that this dataset can be considered really noisy since there are a lot of unfinished contours on the images (Figure 3). For example, there is very often no marking of dirt roads on the pictures with asphalted roads, although in other pictures such dirt roads are highlighted. This causes a large number of problems and significantly worsens the result of the work of the neural network. The worst thing is that we do not know in what form the images are presented in the final sample. Therefore, we can not delete invalid data and must leave all the work of finding roads for the neural network.

### 4.2. Quality Control

To measure the quality of a neural network, the Jaccard Similarity Index or IoU (Intersection over Union) is used. This metric is remarkable for characterizing the quality of object recognition in the task of semantic segmentation, taking into account both the pixels mistakenly recognized as true, and the pixels mistakenly recognized as incorrect.

### 4.3. Training process

The dataset was augmented for 1 million of 256\*256 images and was split into train and validation datasets in the ratio of 90/10. We trained several epochs in total with training batch of 10 images on GTX 1080ti. Then we trained several epochs more with same weights on the original dataset with one image per batch. The starting learning rate was 10e-5, and after every epoch, it was decreasing by half. We've got 65% of IoU metric on local validation dataset. The result of the model at the final version was 55.80%.

## 5. Postprocessing

The postprocessing is a rather variable stage of obtaining the result. It includes certain action steps, each of which can include various hyperparameters. The change in these hyperparameters can lead to a significant change in the result both in the positive and the negative way. We have tried to automate the process of finding the most optimal hyperparameters for obtaining a qualitative result. Owing to the postprocessing algorithm, we managed to improve the result of the final submission from 55.80% to 55.96%.

### 5.1. Roads vectorizing

The main task of postprocessing is to vectorize the roads on the image relative to the probabilities obtained as a re-

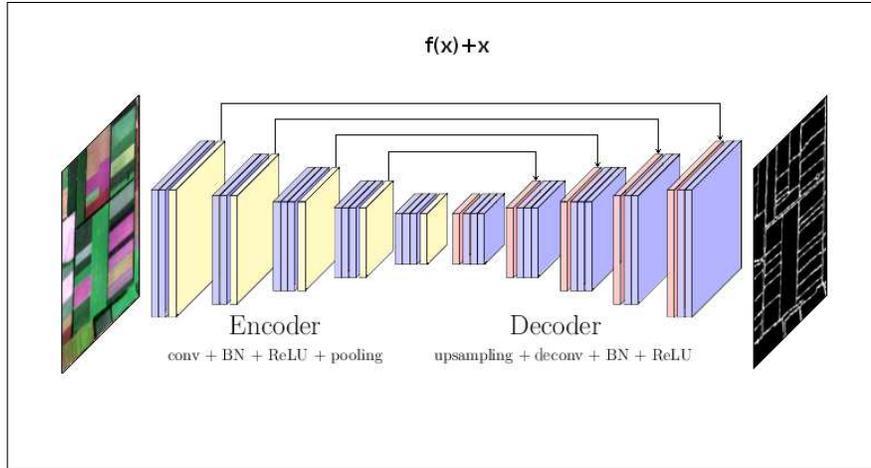


Figure 1. Architecture of Neural Network



Figure 2. Example of images from DeepGlobe Road Extraction Challenge dataset

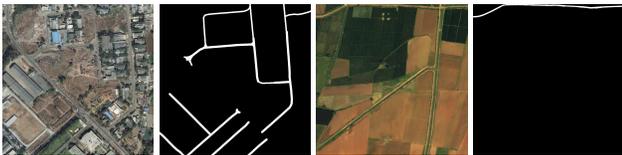


Figure 3. Examples of noising images from DeepGlobe Road Extraction Challenge dataset

result of the work of the neural network (Figure 4). As mentioned before, the road is a straight vector line, which does not significantly change its direction and ends in three cases: abrupt turn of the road, crossroad or collision with the boundaries of the image.

In order to build a vector representation of roads, we decided to clusterize our images roads connected components with the KMeans method using the coordinates of the white points characterizing the presence of the road. In so doing, we've clusterized even those components that consisted of a single pixel. In a number of cases, this approach made it possible to qualitatively finish the disrupted roads. At

this stage there are at once 2 hyperparameters: the value of Threshold for converting a probabilistic image into a binary one and the number of clusters by which roads will be broken to construct vectors for them. We decided to optimize the search for these hyperparameters as follows. In the case of Threshold the number of white pixels in the final image does not exceed 30% and does not go below 0.1%, in the first case the Threshold is decreasing, and in the second increasing. These numbers were taken from the validation dataset and correspond to the maximum and minimum ratio of the number of pixels of the road mask to the pixels of the entire image (Figure 5). As for the number of clusters, we find it according to one rule: the distance between two neighboring clusters should be in the range of 25 to 30 px. These values were derived empirically and give a good result for a narrow road, but in the case of wide roads, the centroids begin to be arranged in two rows. This became one of the main reasons for leveling roads relatively to the middle (Figure 6).

The next step is to connect all possible combinations of points. For this task, two more hyperparameters appear: "brightness of the pixels in the drawn line" and "distance between centroids". The line will be retained if its corresponding indicators exceed the first parameter and will not be greater than the second. Thus, all of the possible road routes will be formed.

Then it necessary to form the road vectors from the constructed routes such way that there remains only one line leading to the nearest centroid in the radius of  $45^\circ$ . Thus, there are only short, neat connections between centroids that characterize the chains of roads.

The final stage is dividing the chain of roads into separate direct vectors by searching for crossroads and sharp turns of the road. Thus forming full-fledged independent roads on each image.

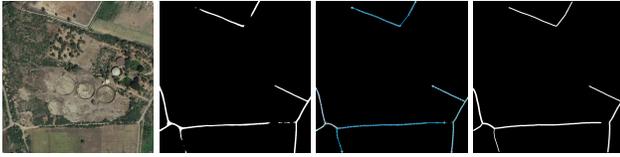


Figure 4. Example of vectorized roads above the image from the neural network



Figure 5. Maximum and minimum ratio of roads mask to all pixels of image

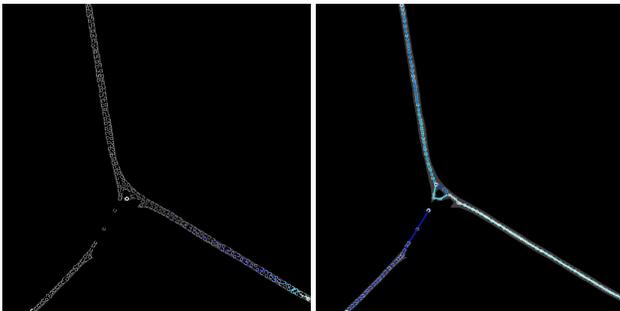


Figure 6. Example of vectors aligning during the road

At this point the basic task of postprocessing - the vectorization of the image roads - is completed. These vectors are already sufficient for a large range of manipulations over them and for improving the results of classification.

## 5.2. Determining a single road width

After separating the roads from each other it is possible to calculate its average width. This is a reasoned approach since the road often has the same width across its entire length. The approach of filling the road with a single-width will help us to fill the gaps that were left by the neural network and get rid of unnecessary spots that the network found by accident. The calculation is made using the maximum radius of the inscribed circle for each centroid. After that, the average resulting width is applied over the mask. Alternatively, it is possible to impose a width simply over the vector, without taking into account the mask obtained from the neural network. Another option for calculating the width and filling with a single-width can be done relative to each pixel on the road line. Also, you can determine the width relative to the centroids and fill in - relative to the pixels of the road line and vice versa.



Figure 7. Example of prolongation of the interrupted roads and removing bad roads (green - added pixels, red - removed pixels)

## 5.3. Prolongation of the interrupted roads

If a certain vector of the road suddenly broke off and thus formed two different roads, there is a possibility of finishing them (Figure 7). All you need to do, is to make sure that the next centroid lies at a distance less than defined hyperparameter, and the direction of the two centroids that you want to connect does not deviate by more than  $5-10^\circ$  or another angle you'll find applicable. In this case, you can say with a great certainty that this is one road, randomly broken by the neural network.

## 5.4. Bad roads removing

All the roads the ends of which do not connect with other roads (by turning or meeting at crossroads) and do not go through the image's borders consider as bad roads (Figure 7) in this context. That is, all the roads that hang in the middle of the image will be deleted. It's easy to get rid of such roads by searching and removing all the clusters, the centroids of which are included in the vector of a bad road.

## 6. Conclusion

The result of the work done was a competitive neural network based on the UNet architecture with the addition of a number of modifications that significantly improve the roads segmentation result. In addition, we defined a number of rules and characteristics of roads, so that from the output we've got from the neural network we could extract and characterize individual roads, get rid of unnecessary noise and try to restore routes that were mistakenly ignored by the neural network. This result can be used to build a single road vector and to build real road masks, taking into account the identified statistics about roads and their characteristics, as well as for more specific tasks. As a part of DeepGlobe Road Extraction Challenge we have reached a result of 55.96% of IoU metric on final dataset.

## References

- [1] V. Badrinarayanan, A. Kendall, and R. Cipolla. Segnet: A deep convolutional encoder-decoder architecture for image segmentation. October 2016. arXiv:1511.00561v3.
- [2] A. Chaurasia and E. Culurciello. Linknet: Exploiting encoder representations for efficient semantic segmentation, June 2017. arXiv:1707.03718v1.

- [3] I. Demir, F. Hughes, A. Raj, K. Dhruv, S. Muddala, S. Garg, B. Doo, and R. Raskar. Generative street addresses from satellite imagery. *ISPRS International Journal of Geo-Information*, 84(7):1–22, March 2018.
- [4] I. Demir, K. Koperski, D. Lindenbaum, G. Pang, J. Huang, S. Basu, F. Hughes, D. Tuia, and R. Raskar. Deepglobe 2018: A challenge to parse the earth through satellite images, 2018. arXiv:1805.06561.
- [5] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. December 2015. arXiv:1512.03385v1.
- [6] V. Iglovikov and A. Shvets. Terausnet: U-net with vgg11 encoder pre-trained on imagenet for image segmentation, January 2018. arXiv:1801.05746v1.
- [7] I. Kahraman, M. K. Turan, and I. R. Karas. Road detection from high satellite images using neural networks. *International Journal of Modeling and Optimization*, 5(4):304–307, August 2015.
- [8] G. Mattyus, S. Wang, S. Fidler, and R. Urtasun. Enhancing road maps by parsing aerial images around the world, 2015. ICCV '15 Proceedings of the 2015 IEEE International Conference on Computer Vision (ICCV).
- [9] M. A. Rahman and Y. Wang. Optimizing intersection-over-union in deep neural networks for image segmentation, 2016.
- [10] O. Ronneberger, P. Fischer, and T. Brox. U-net: Convolutional networks for biomedical image segmentation. May 2015. arXiv:1505.04597v1.
- [11] S. S. Sehra, J. Singh, and H. S. Rai. Assessment of open-streetmap data - a review. *International Journal of Computer Applications*, 76(16):17–20, August 2013.
- [12] Z. Zhang, Q. Liu, Member, and Y. Wang. Road extraction by deep residual u-net, November 2017. arXiv:1711.10684v1.