УДК 681.5.004.4

Bohdan Smishchenko, Artem Volokyta

CREATING METHOD FOR ROAD IMAGE SEGMENTATION

This article examines the method used to generate street markings from existing photographs. The use of neural networks for image generation is demonstrated. The method was implemented using tensorflow based on data from the cityscape dataset.

keyword: cGAN, convolutional neural networks, image generation, data generation.

Fig.: 9, Bibl.: 5

Actuality: This article demonstrates approach of using conditional GAN for segmented image generation. Shows usage of neural network for image generation and teaching neural network for making markup out of image. To implement neural network will use tensorflow as frameworks and cityscape dataset as data source.

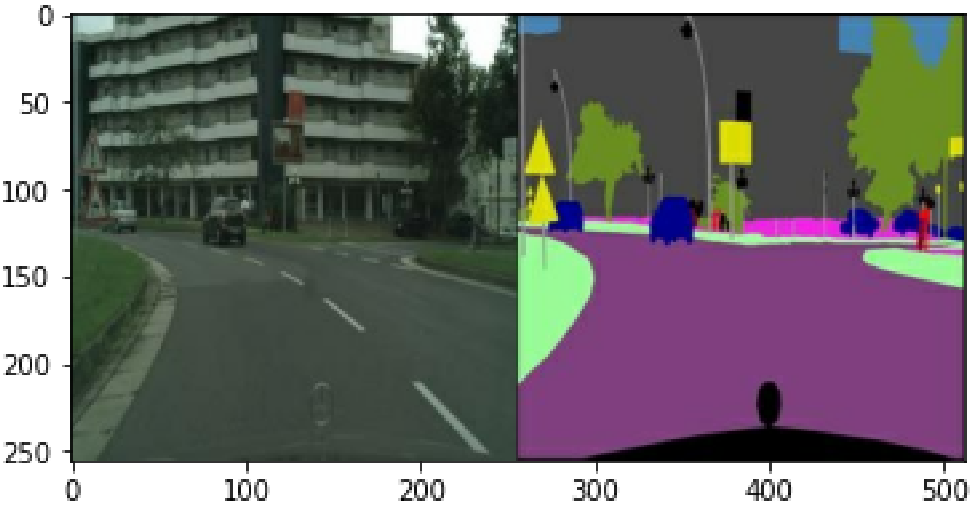


Fig 1. Data example from dataset.

As you can see on Fig 1 we got two images: plain image and segmented one. Idea behind this app is to make ai create layout and learn it to find out patterns inside usual images of streets, nevertheless results are not industry breakthrough and similar task was solved tens of times other ways this approach can be interesting as on of possible applications of this architecture.

Actual scientific research and issues analysis. Methods applied to solve problem of image segmentation usually include usage of convolutional neural networks and deep learning. Multiclass segmentation is one of possible solutions of this task.

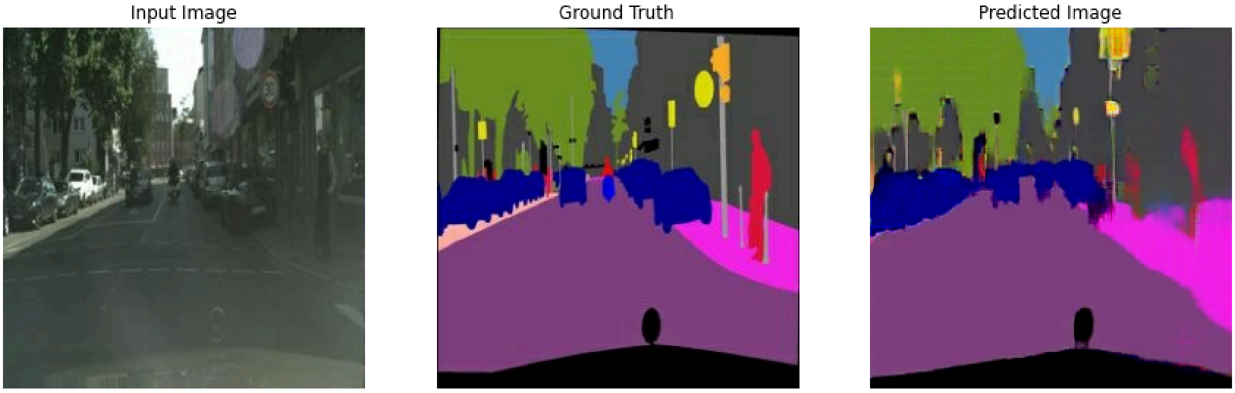


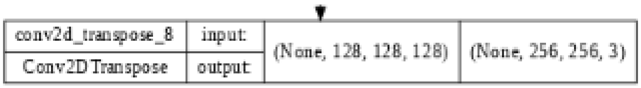
Fig 2. Image segmentation process training result.

Each neural network architecture differently approaches to solution of this problem, GAN trains two networks one for generating image, second for discriminating generated image and real one.

Uninvestigated parts of general matters defining. Despite having multiple solutions that make segmentation images out of street views usage of GANs might offer nice and effective implementations with high accuracy.

The research objective. Purpose of this article is to create method for image segmentation and make research about ways to improve quality of generated images.

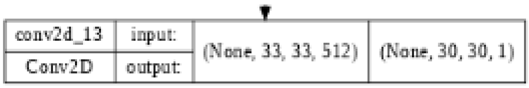
The statement of base material. As input image we got two images 256*256*3 size and result of neural networks are, for generator 256*256*3, for discriminator 30*30*1, where 30 is number of classes inside cityscape dataset.



conv2d_transpose_8	input:	(None, 128, 128, 128)	(None, 256, 256, 3)
Conv2DTranspose	output:		

Fig 3. Generator output.

Let`s see output of discriminator.



conv2d_13	input:	(None, 33, 33, 512)	(None, 30, 30, 1)
Conv2D	output:		

Fig 4. Discriminator output.

After discriminator compares generated image and real one, weights of both are updated and next step of training begins.

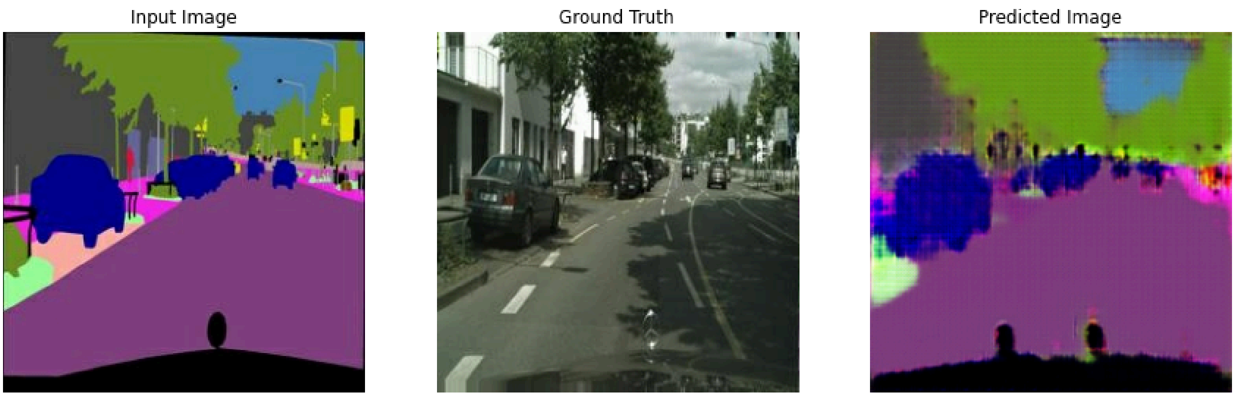


Fig 5. Results of 40000 steps of training on fist image.

After 40000 training steps we got decent results on first taken image, but still requires some clean up, making edges sharper.

Some graphs that show speed of learning neural network, we have two values: discriminator loss and generator loss.

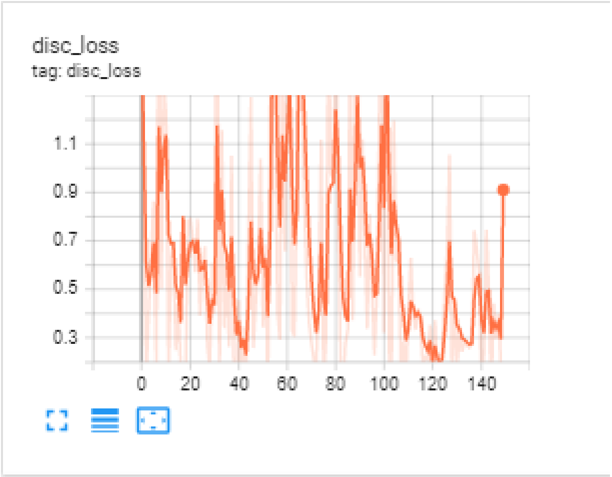


Fig 6. Graph of discriminator loss value changed

Similar graph for generator.

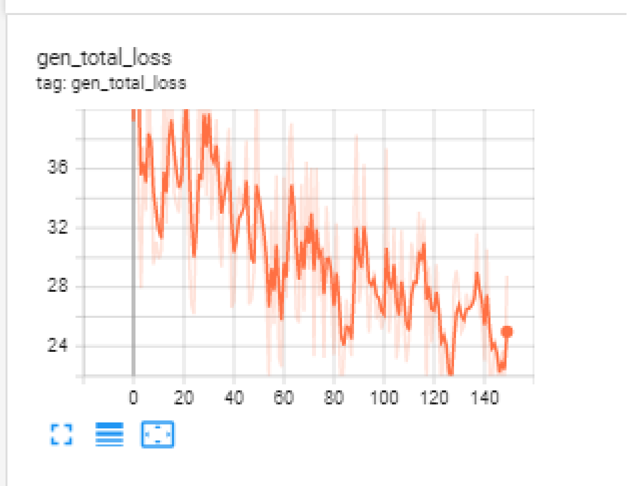


Fig 7. Graph of generator loss value

For generator it`s clearly visible that loss reduces that means neural network has less mistakes over training.

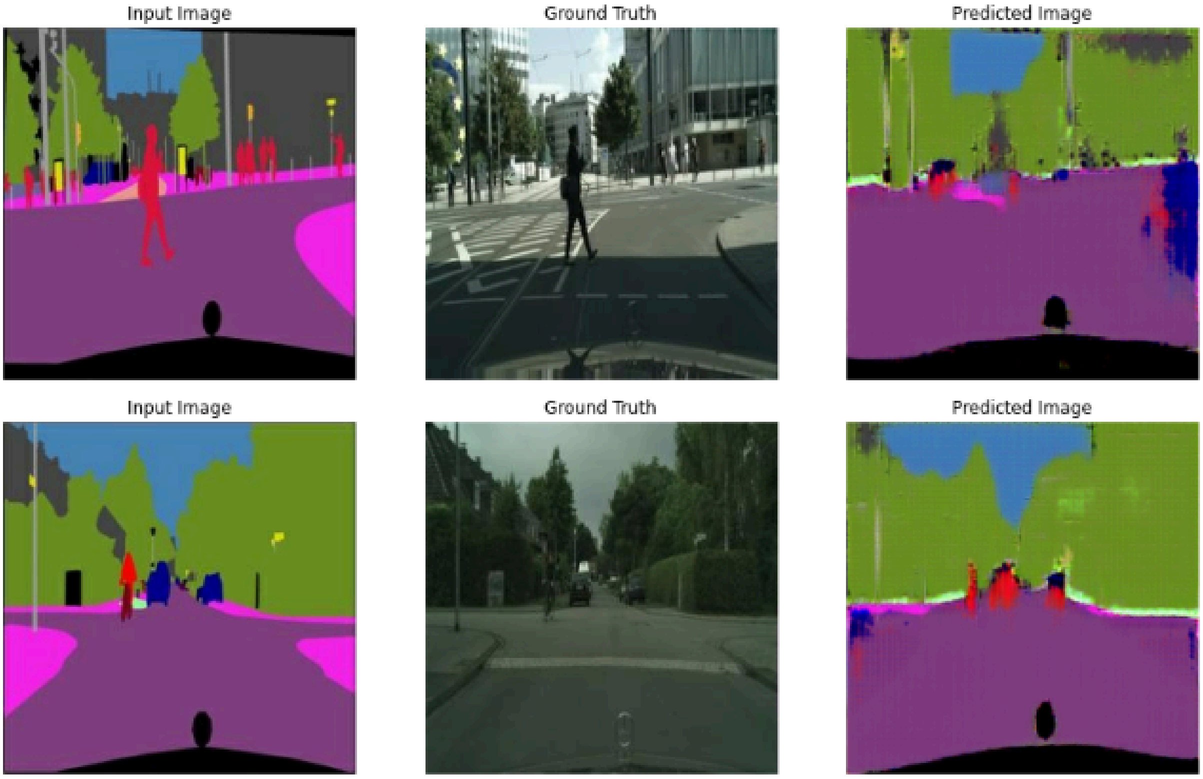


Fig 8. Generated examples on random input from datasource.

Possible ways to improve model. One of the ways to improve quality of generated images is to change architecture of generator or making more steps for training, change amount of classes taking into account, for example reducing number of small details.

Experiments. One possible experiment is to change amount of training steps up to 60 thousand. Result of this is sharper edges of items therefore better quality of generated images.

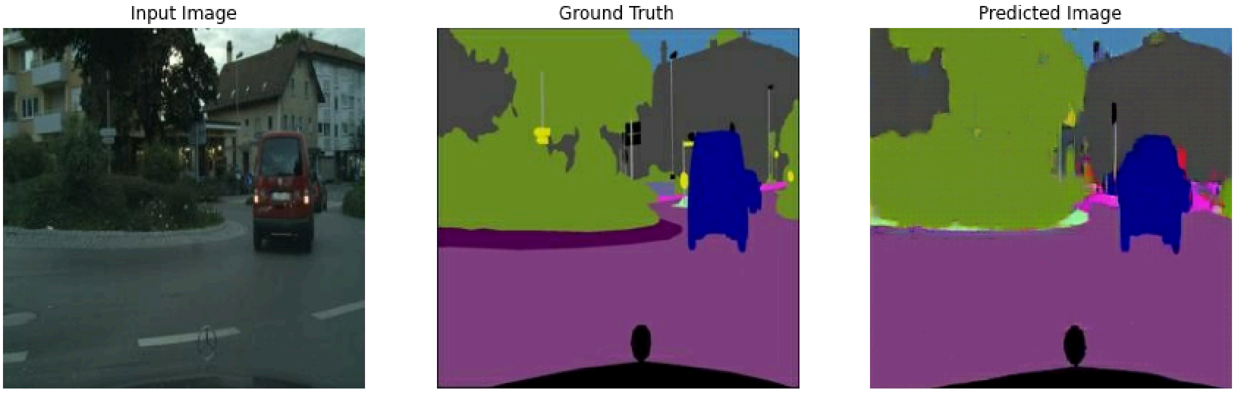


Fig 9. Result of 60000 steps.

Despite huge quality improve this way is limited up to a point where next training steps won`t give any significant quality improvement but required time for training will skyrocket.

Conclusion. In this article was described idea of using this architecture to solve usual problems. In theory described possible ways to improve generated image quality. Provided example of one of such approaches and described it`s drawbacks.

In future works some advance in architecture are main possible approach to make results better and maybe reduce computation time.

References

1. pix2pix: Image-to-image translation with a conditional GAN: https://www.tensorflow.org/tutorials/generative/pix2pix#build_the_discriminator

2. How to Develop a Conditional GAN (cGAN) From Scratch: https://www.tensorflow.org/tutorials/generative/pix2pix#build_the_discriminator

3. How to Develop VGG, Inception and ResNet Modules from Scratch in Keras: https://machinelearningmastery.com/how-to-implement-major-architecture-innovations-for-convolutional-neural-networks/

4. Aurelien Geron Hands-on Machine Learning with Scikit-Learn, Keras, and TensorFlow: Concepts, Tools, and Techniques to Build Intelligent Systems, 2019 600с.

5. ADVANCING PRODUCT DESIGN WORKFLOWS IN MANUFACTURING: https://www.nvidia.com/content/dam/en-zz/es_em/Solutions/design-visualization/industries/manufacturing/quadro-manufacturing-industry-brochure-us-nvidia-681594-FNL-web.pdf

AUTHORS

Volokyta Artem – associate professor, Department of Computer Engineering, National Technical University of Ukraine "Igor Sikorsky Kyiv Polytechnic Institute".

E-mail: artem.volokita@kpi.ua

Bohdan Smishchenko – student, Department of Computer Engineering, National Technical University of Ukraine "Igor Sikorsky Kyiv Polytechnic Institute".

E-mail: bohdan.smishenko@gmail.com