UDC 004.032.26

S. I. Shapovalova, Yu.V. Moskalenko

SEMANTIC SEGMENTATION ACCURACY IMPROVEMENT BASED ON FORCED EDGE DETECTION

Executive Summary: The paper suggests a method of deep learning networks U-Net and FPN modification to improve pixel accuracy of semantic segmentation. The use of Sobel operator for additional definition of object edges is justified. The efficiency of the proposed method is experimentally proved.

Keywords: deep learning networks, U-Net, FPN, semantic segmentation, Sobel operator.

Introduction

The problem of semantic segmentation is one of the most common problems of computer vision. Examples of such tasks are: detection of objects on the scenes by onboard systems of unmanned aerial vehicles, analysis of medical images (in particular findings and abnormalities diagnostics), recognition of objects on satellite images and others.

The solution to this problem has received a new impulse with the appearance of deep learning networks. The use of these networks has provided fast and sufficiently accurate image analysis. However, in many cases, the solution of the semantic segmentation problem requires pixel accuracy because based on the scale, the image can correspond to real sizes of different orders. For example, in cases of analysis of medical images – it is millimeters, and in objectives of seismic method of prospecting – meters. Basic algorithms of deep learning networks may not be enough for the required accuracy of solving semantic segmentation problems. Therefore, the research to improve the accuracy of semantic segmentation is relevant and has practical significance.

Recent studies analysis

One of the first neural networks designed specifically for segmentation was SegNET [1]. The limitation of SegNET is that when the current image is transferred through the entire network, the pixel accuracy is lost.

U-Net [2] network partially solves this problem. The architecture of this network, like SegNet, consists of two parts: left - encoder, right - decoder. A distinctive feature of the network is that all blocks are connected not only inconsistently, but also by skip-connections. Skip-connections copy the output of the encoder layer to the corresponding decoder layer.

Similar in concept to U-Net network is Feature Pyramid Network (FPN) [3]. The difference, firstly, is that the output of each encoder block is not copied to the corresponding decoder block, but passes through the layer. Secondly, the U-Net makes predictions once at the end of the decoder. And the FPN network makes predictions on each decoder block. After that, the generalized prediction is performed by scaling all previous predictions of the same size with the final convolution.

Another way to significantly improve the quality of segmentation is to replace the encoder with an Autonomous network backbone [4]. Such networks are most often previously trained on certain datasets. The speed of learning and the quality of network segmentation based on a pre-trained encoder is usually better because the backbone network already has the property of highlighting certain features. Trained backbone networks are included in popular neural network libraries such as MXNet [5], Keras [6], TensorFlow [7], PyTorch [8].

Attention mechanisms have also become common means of improving the accuracy of results on deep learning networks [9, 10].

Approbation of modern improvements of deep learning networks is performed both on practical tasks and at special competitions, in particular on the Kaggle platform [11]. A distinctive feature is that the solution that determined the winner on the current task is often not optimal for the other. However, all means and methods of semantic segmentation optimization are accumulated to solve the following problems. At the competitions on problems solution, as a rule, it is estimated by measures according to the results of which the winner defines 4-5 signs after a decimal point. Therefore, further research is necessary to improve deep learning networks to increase the accuracy of semantic segmentation.

Purpose and objectives of the study

Purpose: to improve mathematic and algorithm provision of the deep learning networks U-Net and FPN modification to improve pixel accuracy of semantic segmentation. For this purpose the **tasks** are solved:

1. Definition of forced discovery algorithm of the edges on the image.
2. Architecture modification of U-Net and FPN networks in the proposed way.
3. The conduct of computational experiments to solve semantic segmentation problems for comparative analysis of the results obtained on basic and modified networks.

The detection of image edges based on the Sobel operator

The segmentation quality increase can be improved at the architecture level by modifying existing blocks and/or creating blocks with new functionality. When training convolutional networks, convolutional filters also highlight edges. However, after passing through many convolution layers on the lower layers of the encoder, the edges of the object become blurred. In this paper the assumption that in applied objectives often the edges of the object can be found behind the gradient change is used. Since segmentation tasks we need to know the edges of the object exactly, additional edges highlighting should improve network performance.

Edges detection in the image is possible with the help of Kenny, Roberts, Pruitt, Sobel operators. The study [12] analyzes the effectiveness of these operators implementation. The use of Kenny and Sobel operators is recognized as the best in accuracy. However, the Sobel operator has advantages such as ease of implementation affecting the processing speed and easy integration with convolutional neural networks.

For convolution of the input image the Sobel operator uses 2 matrices:

$$Sx = \begin{pmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ 1 & 2 & 1 \end{pmatrix}; \ Sy = \begin{pmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{pmatrix}. \tag{1}$$

The use of matrices (1) effectively highlights the edges of those objects the edges of which are close to the X axis (the X axis is directed to the right) or the Y axis. Additionally, you can use convolution matrices of the form:

$$Sd_1 = \begin{pmatrix} 2 & 1 & 0 \\ 1 & 0 & -1 \\ 0 & -1 & -2 \end{pmatrix}; \ Sd_2 = \begin{pmatrix} 0 & 1 & 2 \\ -1 & 0 & 1 \\ -2 & -1 & 0 \end{pmatrix}. \tag{2}$$

Matrices (2) allow efficient selection of the edges close to diagonals.

Thus, you can define the following convolution operations:

$$\begin{aligned} Gx &= Sx * A, \\ Gy &= Sy * A, \\ GD_1 &= Sd_1 * A, \\ GD_2 &= Sd_2 * A, \end{aligned} \tag{3}$$

where Gx, Gy, Gd1, Gd2 — the results of the convolution along the axes and diagonals,

Sx, Sy, Sd1, Sd2 — Sobel matrices (1), (2),

A - the input image transmitted as a matrix of pixel values.

The resulting image is obtained by combining the results:

$$G = \frac{1}{2}\sqrt{Gy^2 + Gx^2 + Gd_1^2 + Gd_2^2}\ , \tag{4}$$

where G - the final image;

Gy, Gx, Gd1, Gd2 — the results of the convolution along the axes and diagonals,

The results of applying the concatenation of all component mappings to the Sobel operator with the direct result of its calculations (4), gives better results than using only the Sobel operator.

As a result, after applying all operations, such a set of matrices is obtained:

$$Out = [G, Gy, Gx, Gd_1, Gd_2]\,. \tag{5}$$

Further, the implementation of **Out** calculations will be called the Sobel block.

As a rule, input images are not single-channel images (black-and-white), but two- and multi-channel images (for example RGB-images). The Sobel operator is applied for each channel independently. For example, for an RGB image, 15 channels will be received, which will be reproduced by a set of matrices OutR, OutG, OutB.

Architecture modification of deep learning networks

In this paper, the research was carried out on U-Net and FPN networks. The elementary layers of these networks are convolution, activation, concatenation, pooling, addition, UpSampling.

To represent the modified architecture, we introduce the concept of a block, firstly noting that the dimension of the image is generally defined as:

$$Shape = W * H * C, \tag{6}$$

where G - the image height; H — image width, C — number of channels of the image.

Let's define a block as a group of elementary layers of a neural network processing the input data of fixed dimension only in width (W) and height (H).

Figure 1 represents the block architecture of a modified UNet network where the Sobel operator is applied.

In the figure, the repetition of the blue block emphasizes the data copying from the output of the encoder block to the corresponding decoder block. The raw input image is given to the encoder and then goes through standard processing. In the proposed architecture, only a few upper (last) decoder blocks are modified. Figure 1 shows that these blocks have added edges forced discovery (highlighted in yellow) and their concatenation with the corresponding encoder blocks. Studies have shown that the best results are obtained by using the proposed method on the last 2-3 blocks.

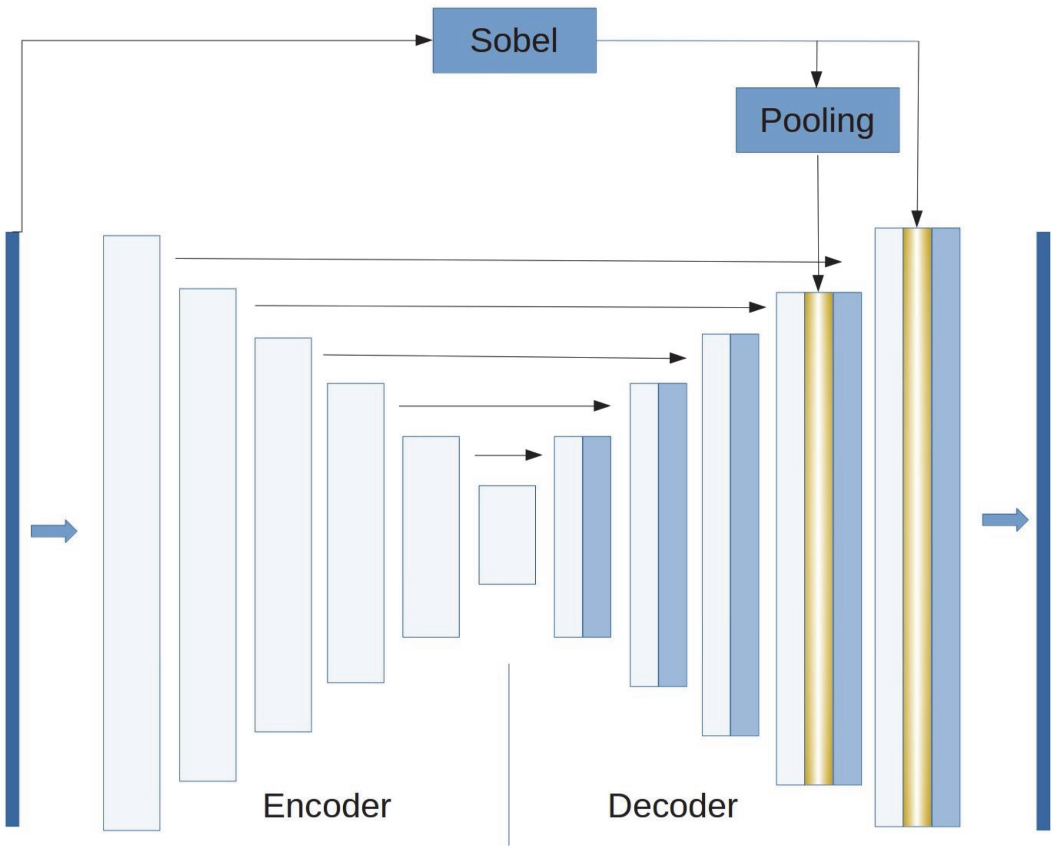

Figure 1.U-Net network upgrade

A similar conversion is used in the FPN network. The difference is that the result is combined with the output of the decoder block and subsequently is transmitted unchanged to the convolution layer.

Since the dimensions of the layers do not match, it is necessary to reduce the result (5) to the appropriate dimension. The network decoder block has the following dimension:

$$\text{Shape} = (W/k;\ H/k\ ), \tag{7}$$

where W, H - the width and height of the input image, respectively, k – block number from the end.

The pooling layer reduces the output dimension of the Sobel block by k times.

Network modification is implemented in Python on the Keras framework using the segmentation_models library [13].

Computational experiments

Computational experiments were conducted to solve such objectives:

1. Salt identification Challenge [14] - the task of identifying salt deposits in the Earth's interior.

2. Cityscapes [15] - recognition of street scenes of specified objects from standard classes.

The first challenge is to identify salt gaps in the Earth's interior, proposed by the international Kaggle community. This is a binary segmentation task, i.e. you need to determine whether salt deposits are present in seismic images. The dataset contains 4000 single-channel images with a dimension of 101x101 pixels. Before presenting the neural network, all examples were augmented to a dimension of 128x128. The dataset was divided in the proportion of 80: 20 (80% - training sample, 20% - validation).

Cityscape refers to the classical segmentation objectives of the specified objects in photographs of street scenes. The data set being received from the resource [16] contains 2975 images in the training sample and 500 images in the validation one. Segmentation is executed by 13 classes, such as car, pedestrian, road sign and others. The image dimension 256x256 pixels was reduced to 128x128 pixels before being fed to the neural network.

The accuracy metrics in all experiments was the Intersection over Union - IoU (another name – Jaccard similarity coefficient):

$$IoU = AreaOfOverlap / AreaOfUnion, \qquad (6)$$

where AreaOfOverlap - the sectional area of the original and provided area,

AreaOfUnion – the area of association of original and the provided area,

Binary cross entropy is chosen as the loss function for the first objective; for the second – it is selected the categorical cross entropy. The initial learning ratio is 0.002. During the training, the automatic reduction of the learning speed rate coefficient by 5 times was used, if the IoU on the validation sample did not decrease for 5 stages. The early stop principle was also used, whereby learning was stopped if the network did not improve IOU performance over 15 stages. After completing the training, the scales of the stage at which the largest IoU on the validation sample was achieved were loaded into the network.

During the experiments, the basic and modified by the proposed method of U-Net, UNet-Attention and FPN networks were tested. ResNet34 and ResNet50 are selected as backbone networks. For the second task, backbone-networks were trained on ImageNet.

The training was conducted on a computer with an Nvidia GTX 1080 TI video card. The size of the butch for training received was chosen as 64. From augmentations at training only rotation concerning a vertical axis (horizontal flip) with probability 0.5 was applied. Test time augmentation (TTA) under the assumption of the neural network was not applied.

The following series of computational experiments were carried out:

– on the basic models without modifications;

– on models with the addition of the Sobel operator to the last block;

– on models with the addition of the Sobel operator to the last two blocks;

– on models with the addition of the Sobel operator to the last three blocks;

The results of computational experiments on the Salt Identification Challenge are given in Table 1.

Table 1. **The results of computational experiments on the Salt Identification Challenge**

Architecture		**IoU at validation sample on model:**			
Type	**Backbone**	**Basic**	**Using the Sobel block at**		
			1 layer	**2 layer**	**3 layer**
U-Net	ResNet34	0.80672	0.81241	**0.81372**	0.81268
	ResNet50	0.76080	0.76287	0.76854	**0.76901**
U-Net Attention	ResNet34	0.80821	0.81429	0.81463	**0.81487**
	ResNet50	0.76357	0.86520	**0.86613**	0.86579
FPN	ResNet34	0.79366	0.79922	**0.80470**	0.80240
	ResNet50	0.75942	0.76157	**0.76248**	0.76173

The results of computational experiments on the Cityscapes are given in Table 2.

Table 2. **The results of computational experiments on the Cityscapes**

Architecture		**IoU at validation sample on model:**			
Type	**Backbone**	**Basic**	**Using the Sobel block at**		
			1 layer	**2 layer**	**3 layer**
U-Net	ResNet34	0.42502	0.43308	**0.43603**	0.43519
	ResNet50	0.43106	0.43968	**0.44119**	0.44011
U-Net Attention	ResNet34	0.46703	0.47806	**0.48040**	0.47912
	ResNet50	0.47142	0.47593	0.47684	**0.47761**
FPN	ResNet34	0.50421	0.51145	**0.51474**	0.51320
	ResNet50	0.50944	0.51153	**0.51287**	0.51231

The table shows that the Sobel operator improves segmentation accuracy in any case. It is also obvious that when changing the architecture, it is necessary to determine the optimal number of added Sobel blocks.

Conclusions

1. Modifications of deep learning networks based on forced discovery of the edges by Sobel operator to increase pixel accuracy of semantic segmentation are proposed.
2. A modification of the deep learning architecture of U-Net and FPN in the proposed way is suggested.
3. The efficiency of the proposed method is experimentally proved.

The perspective of further research is the improvement of the proposed method for more accurate edges selection and its application for modification of deep learning networks of PSPNet, DeepLab, LinkNet and other architectures.

REFERENCES

1. V. Badrinarayanan, A. Kendall, and R. Cipolla, "Segnet: A deep convolutional encoder-decoder architecture for image segmentation" arXiv:1511.00561, 2015.
2. Ronneberger, O., Fischer, P., Brox, T.: U-net: Convolutional networks for biomedical image segmentation. In: MICCAI. LNCS, vol. 9351, pp. 234–241. Springer (2015)
3. T.-Y. Lin, P. Dollár, R. Girshick, K. He, B. Hariharan, and S. Belongie. Feature pyramid networks for object detection. arXiv preprint arXiv:1612.03144, 2016. 3
4. Iglovikov V., Shvets A. TernausNet: U-Net with VGG11 Encoder Pre-Trained on ImageNet for Image Segmentation. arXiv: 1801.05746, 2018
5. Apache MXNet. URL: https://mxnet.apache.org
6. Keras. URL: https://keras.io/
7. TensorFlow. URL: https://www.tensorflow.org/
8. PyTorch. URL: https://pytorch.org/
9. Hu J., Shen L., Sun G. Squeeze-and-Excitation Networks. arXiv: Computer Sience. 2017. 5 September. 11 p. URL: https://arxiv.org/pdf/1709.01507v1.pdf
10. O. Oktay, J. Schlemper, L. L. Folgoc, M. Lee, M. Heinrich, K. Misawa, K. Mori, S. McDonagh, N. Y. Hammerla, B. Kainz et al., "Attention U-Net: Learning Where to Look for the Pancreas," arXiv preprint arXiv:1804.03999, 2018.
11. Kaggle. URL: https://www.kaggle.com/
12. S. Das, "Comparison of Various Edge Detection Technique", International Journal of Signal Processing. Image Processing and Pattern Recognition, vol. 9, no. 2, (2016), pp. 143-158.

13. Segmentation Models. URL: https://github.com/qubvel/segmentation_models

14. TGS Salt Identification Challenge. Segment salt deposits beneath the Earth's surface. URL: https://www.kaggle.com/c/tgs-salt-identification-challenge

15. Cityscapes Dataset. URL: https://www.cityscapes-dataset.com/

16. Cityscapes Image Pairs. URL: http://www.kaggle.com/dansbecker/cityscapes-image-pairs