Segmentation of analogue meter readings using neural networks

Vadym Slyusar [1], Ihor Sliusar [2], Nataliia Bihun [1], and Volodymyr Piliuhin [2]

[1] Central Research Institute of Armaments and Military Equipment of Armed Forces of Ukraine, Povitrophlotsky Av., 28B, Kyiv, 03049, Ukraine
[2] Poltava State Agrarian University, str. G. Skovorody, 1/3, Poltava, 36003, Ukraine

Abstract
The report discusses options for solving the image segmentation problem of displaying digital indicators of analogue water or gas meters using neural networks. The results of a comparative analysis of the application of various implementation options for neural networks based on PSP, U-Net, and U-Net^2 are presented. The Water Meters Dataset, which is freely available on the Kaggle website, was used as a dataset. In this case, the analysis was carried out by comparing various parameters of the learning process, as well as the value of the accuracy indicator on the validation sample. Its maximum value was reached at the level of 86.5% when using the PSPBlock2D neural network and 88.8% - on the light version of U-Net.

Keywords 1
Neural Network, segmentation, U-Net, PSP

1. Introduction

As you know, one of the constraints to the implementation of the concepts of Smart Home, Smart City, Industry 4.0, IoT, Agriculture 4.0, etc. is the need to integrate analog energy metering, for example, water, gas, and electricity meters. Often their replacement with digital devices is not cost-effective [1]. This may be due to the ban on making changes to communications, the high cost of developing design and technical documentation, a large number of analogue accounting tools at the enterprise, etc.

One option to overcome this barrier could be a combination of artificial intelligence and Internet of Things (AI + IoT) technologies. At the same time, an optical channel for digitizing readings based on the recognition operation is very often used to transform analogue meters into digital ones. Solutions [2] and [3] should be mentioned as an example of such an approach. However, they have significant drawbacks that affect mass adoption. Here it is necessary to indicate high requirements for the spatial stability of the image and exposure parameters, the lack of unification by types of counters and fonts, and the influence of vibrations with certain specifics of technological production processes. In addition, the use of edge computing has to adjust to limited computing resources, e.g. low-resolution images (typically around 28x28), the need for manual segmentation, accurate initial set-up, and manual correction of reading data.

One of the options for solving this problem can be the use of an image segmentation procedure before recognition. The technical implementation of this approach is possible due to fog computing technologies.

Today, to perform segmentation, several approaches are used, considered, for example, in [4, 5].

2. The aim of the research

The aim of the work is a comparative analysis of the accuracy of possible options for solving the problem of semantic segmentation of images of digital displays of counters based on neural networks.

3. The Main Results of the Study

As you know, the key point in the application of neural networks is the choice of dataset. In the field of counters segmentation, this problem is made easier by the fact that the corresponding dataset is publicly available on the Kaggle the relevant dataset is publicly available on the Kaggle website [6].

Since a common option for solving the problem of semantic image segmentation is the use of the PSP neural network [7], we will consider this approach as a starting point for research, using the architecture of the so-called large PSP, shown in Figure 1. Since the indicated structure of the neural network assumes a 16-fold reduction in the data matrix, the image format used for learning should be completely divisible by 16. For this reason, the size of the original images in the Water Meters dataset [6] of 1000x1778 pixels was preliminarily recompressed into sizes that are a multiple of 16.
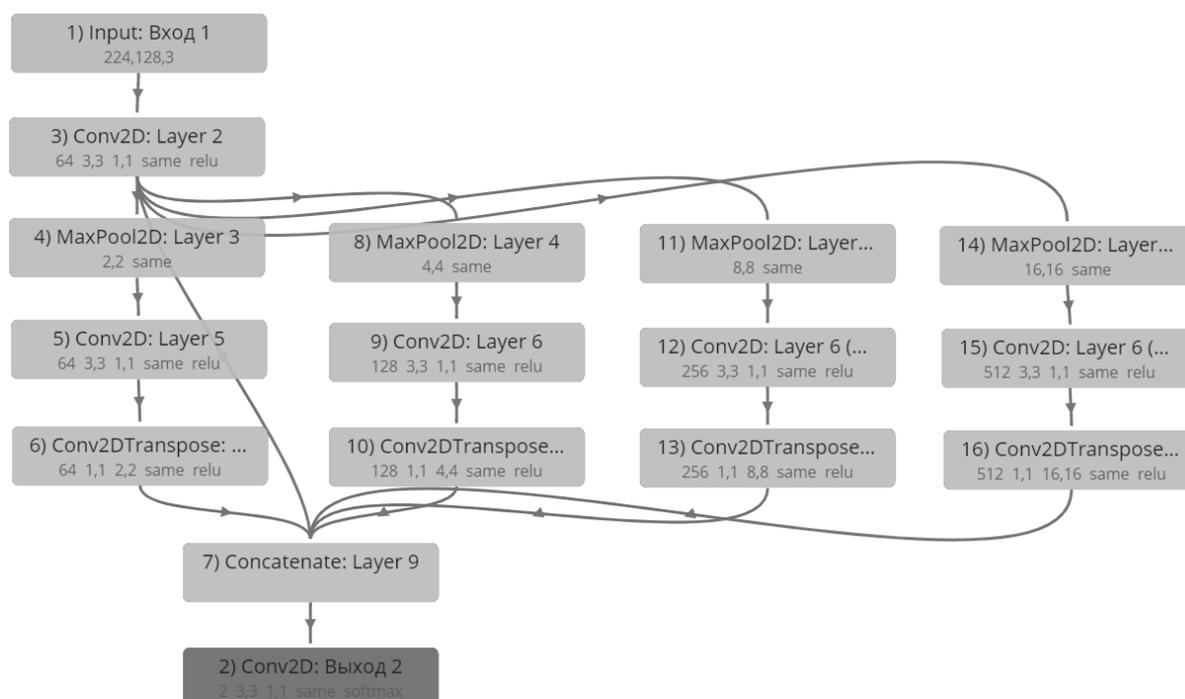
Figure 1: 4-channel PSP

As a first step in solving the learning problem, the 128x224 pixel format was chosen as the closest in proportion to the original photos. In particular, for an image with a frame side of 128 pixels, a recalculation with a factor of 1778/1000 gives a result of 227.584. In this case, rounding to 224 pixels should be almost invisible. Alternatively, with twice the length of the shorter side of the frame of 256 pixels, multiplying by 1778/1000 gives 455.168. In this variant, the closest multiples of 16 would be 464 or 448 pixels. Similarly, at 240 pixels we get:

$$240(1778/1000) = 426.72,$$

with the corresponding nearest multiple of 16 being 432.

It should be noted that the choice of one or another format of the recompressed image, in addition to the maximum possibility of maintaining the proportions of the original images of the dataset, should also take into account the existing limitations of the computing resources on which the neural

network is trained. To maximise these resources, the study used the graphical mapping capabilities of Google's ColabPro+ service in learning.

The learning process of the PSP neural network was performed with a learning step of 0.001 and batch 16 since the installation of batch 32 was accompanied by an insufficient resource error. The learning dataset contained 870 images, and the validation dataset contained 374 images. The masks used for image segmentation were black and white. The percentage of space occupied by the black background was 98%, while the white cutout for the digital display accounted for 2%. The run time for 200 learning epochs in standard Google ColabPro+ connection mode with a V100 graphics card equipped with 16 GB of RAM was 32 minutes. At the same time, the maximum learning accuracy on the original large PSP reached 70% at the 73rd epoch. Continuing learning to 400 epochs made it possible to achieve an accuracy of 75.2% at the 356th epoch.

Next, we explored the option of modifying the original PSP architecture by replacing the Conv2DTranspose layers with UpSampling and MaxPooling with AveragePooling (Figure 2). The obtained results of the validation of such a neural network after learning to allow us to conclude that the modification of the original PSP architecture with Conv2DTranspose layers on the described dataset is trained worse than the modification with PSP with UpSampling layers. In particular, the modified version made it possible to obtain an average class accuracy of 81.1% already at the 48th epoch.
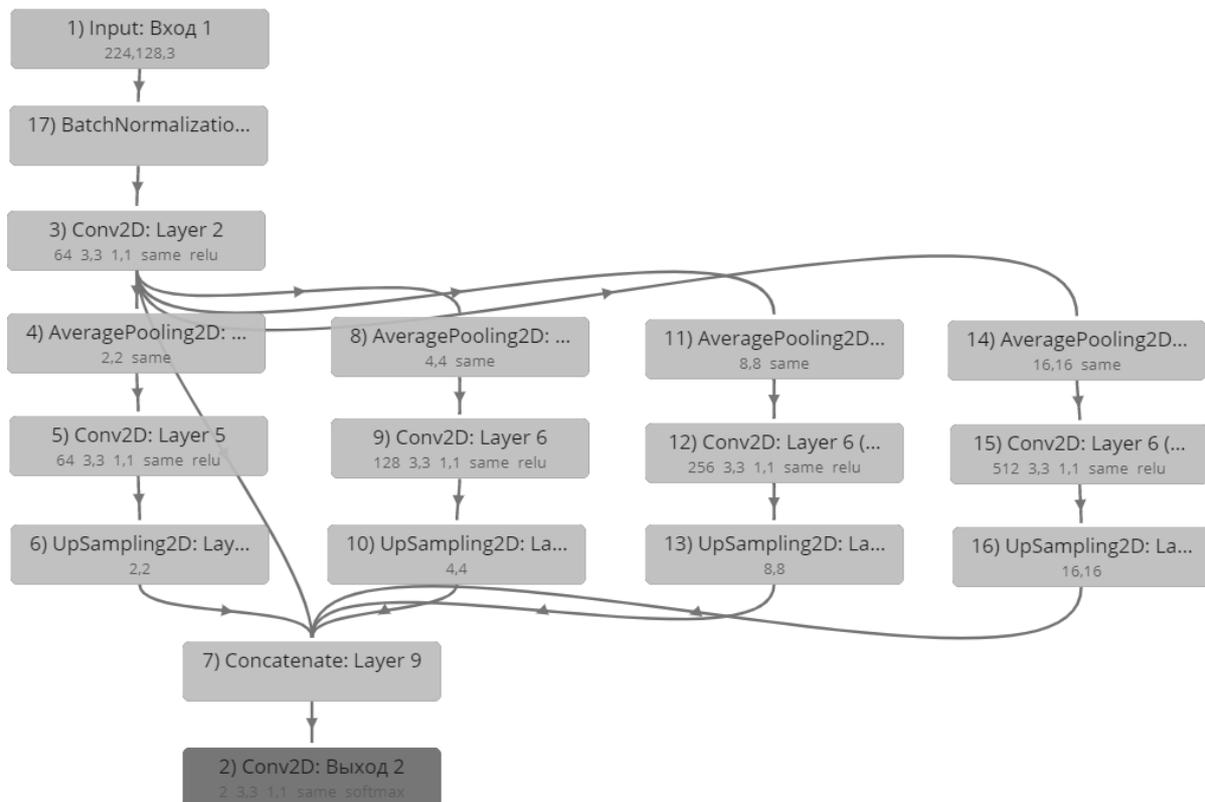
Figure 2: Modification of the original PSP neural network

Even greater improvements in accuracy were achieved by using an alternative to Figure 2 modification of the PSP-network of Figure 1, which consisted in increasing the number of convolutional layers with the ReLu activation function in each of the channels up to 8. The size of their cores remained the same (3x3), fixed for all 4 channels (Pooling branches) also remained and the number of convolution kernels is 16. In this case, batch normalization layers were used in each channel and, additionally, a Dropout layer with a data thinning factor of 0.1 was applied at the channel output. This structure has been given the conventional name PSPBlock2D (Figure 3).

As a result of learning for 160 epochs, the accuracy of the class with the worst segmentation quality on the test sample reached 73%, and the average class accuracy reached 86.2%. In Figure 4 illustrates the learning process described, and Figure 5 – the segmentation quality on the validation set.
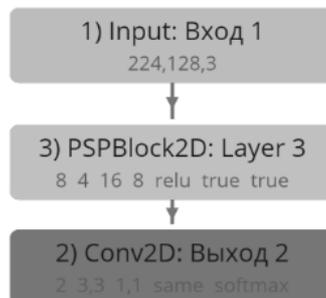
Figure 3: PSPBlock2D neural network structure

Figure 4: PSPBlock2D neural network learning results

At the next stage, the studies were carried out using a neural network of the U-Net type [8 - 12].

In Figure 6 shows the architecture of the light version of U-Net. The relative simplicity of its architecture made it possible to switch to the original 432x240 pixel dataset and easily carry out long-term learning for 622 epochs with a batch of 16 and a final learning step of 0.00001. The calculation time of one epoch fluctuated within 27 - 28 sec. Already at the 54th epoch, an accuracy of 88.4% was achieved, and then it took more than 400 epochs for the maximum achieved accuracy to stabilize at the level of 88.8% at the 464th epoch. Not only the architecture of the neural network contributed to the improvement in accuracy, but also the larger image format during learning.

This can be confirmed by the results of using the more complex structure of the so-called medium U-Net, schematically shown in Figure 7, with a learning image format of 224x128 pixels. At the same time, an accuracy of 87.8% was achieved at the 49th epoch with batch 32.

The architecture of this neural network included 5 serially connected base blocks CB (Figure 8) in the descending branch and 4 base blocks IB (Figure 9) in the ascending branch.
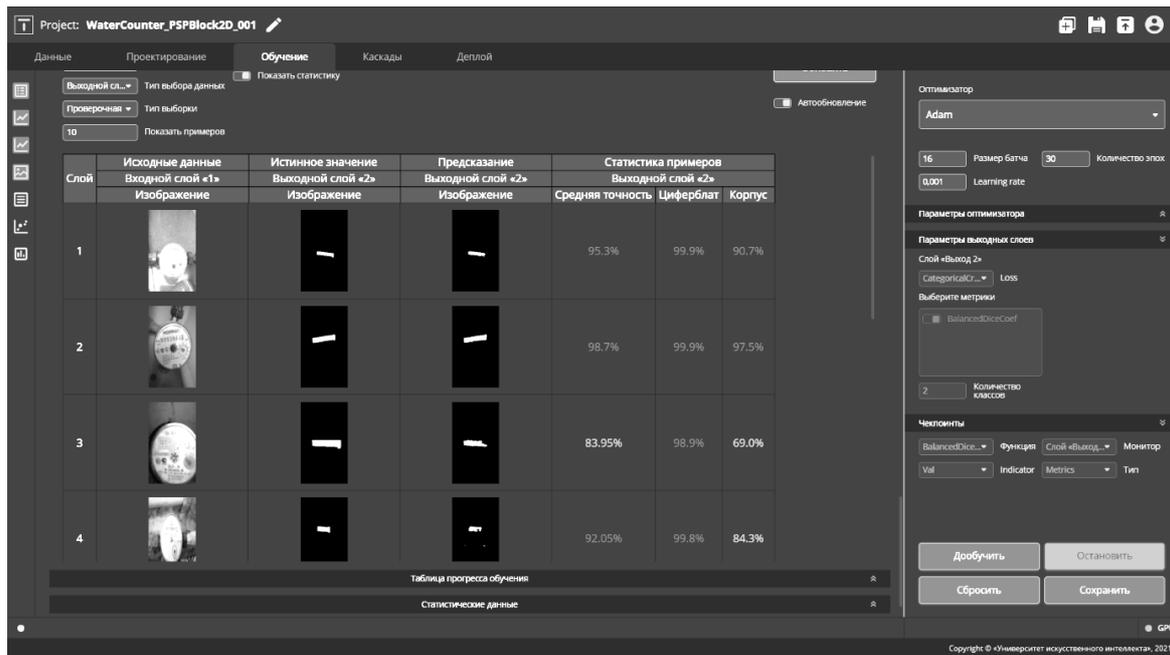
Figure 5: PSPBlock2D neural network learning results
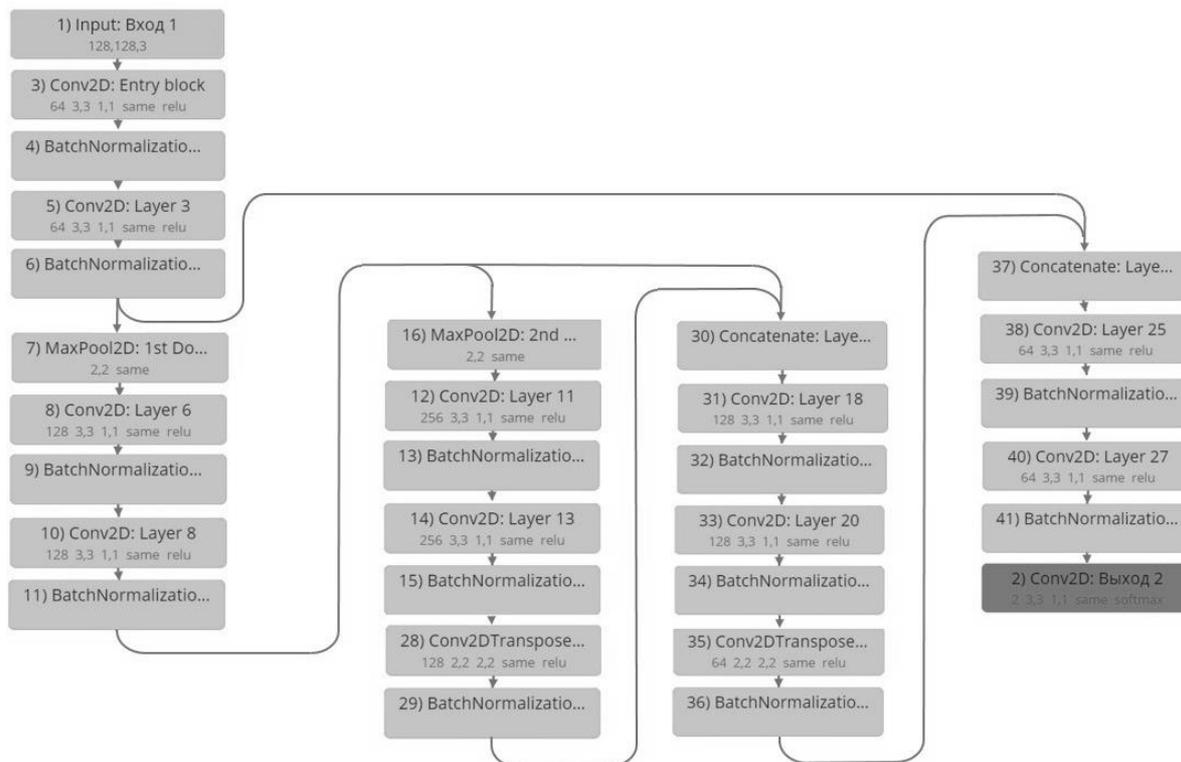
Figure 6: Lite version of U-Net

At the same time, inside each base block of both branches, the same type of Conv2D and Conv2Dtranspose convolutions with ReLu activation functions, respectively, were used. However, the number of convolution cores increased from 16 to 256 multiples of degree 2 in the descending branch and decreased in the opposite order, from 128 to 16, in the ascending branch when moving from one base unit to another.

In the MaxPool2D layers the Pool size = 2,2. The Quantity of filters in Conv2D layers of CB_m blocks is $K=2^{3+m}$, m=1, …, 5, the Kernel size is 3x3 and strides – 1x1, Padding = same, Activation function is ReLu.
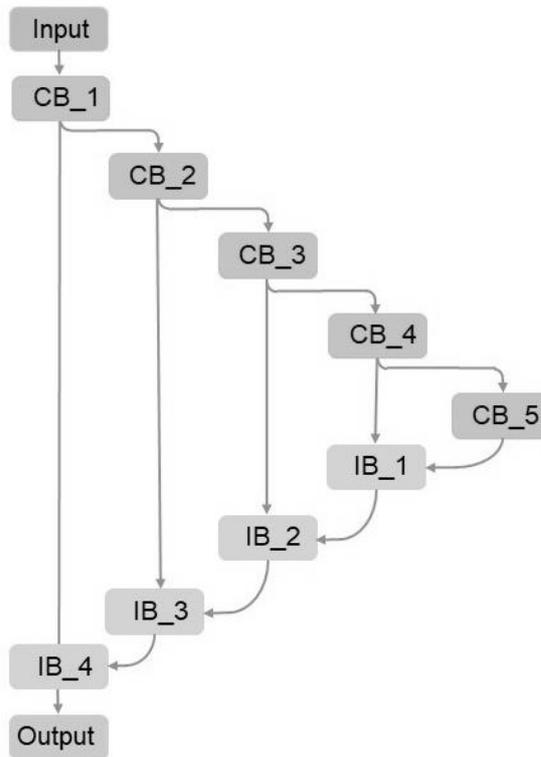
Figure 7: Modified version of U-Net medium architecture.
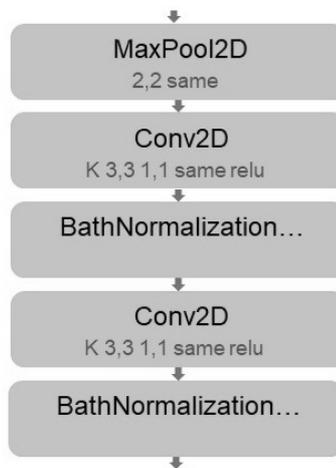
Figure 8: Typical downstream middle U-Net building block (CB)

The Quantity of filters in Conv2DTranspose and Conv2D layers of IB_r blocks is $L=2^{8-r}$, r=1, …, 4. The Kernel size of Conv2D is 3x3 and strides – 1x1, Padding = same, Activation function is ReLu. The Kernel size of Conv2DTranspose is 2x2 and strides – 2x2.

We also studied a neural network variant similar in structure, called the large U-Net, which differed by increasing the number of convolution cores in descending blocks by the sequence 64, 128, 256, 512, 1024 and changing them in reverse order in the ascending branch. Contrary to expectations, such a manoeuvre with the parameters of the architecture did not improve the accuracy, which was limited to the level of 87% at the 54th epoch with the same batch sizes (32) and learning step.

A further complication of the architecture was carried out by switching to a neural network of the U-Net++ type (Figure 10). In the course of computational experiments, it was found that this neural network works with batch 4, but not as efficiently as a large and medium U-Net. As you might expect,

batch 8 at 0.001 gives better accuracy than batch 4. Also, U-Net++ works on batch 16, but much worse at 0.001. The resulting accuracy for different batch sizes is presented in Table. 1.
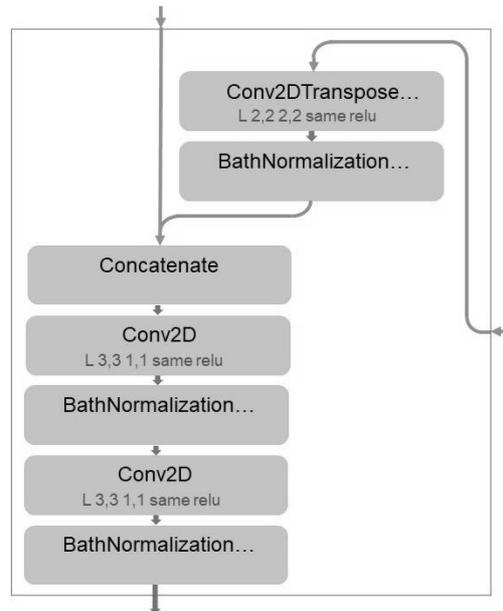
Figure 9: Typical IB building block of the ascending branch of the middle U-Net
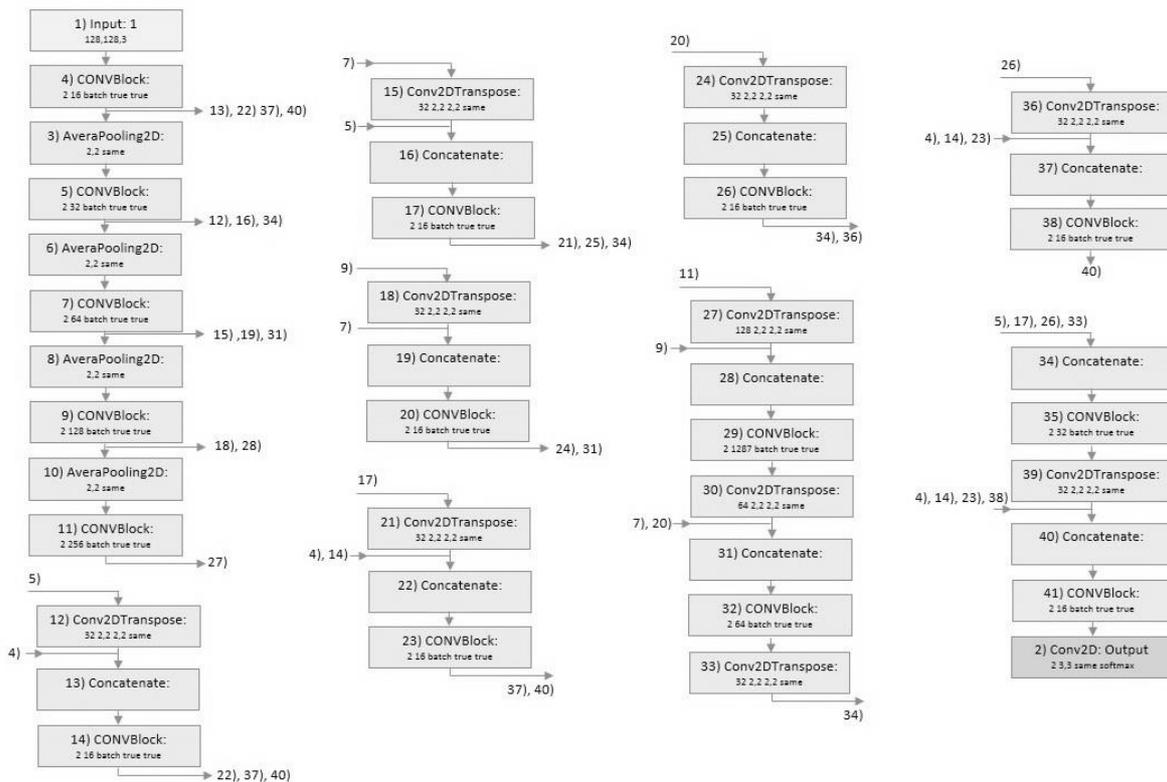
Figure 10: Architecture "U-Net++" from the Terra AI framework

The so-called U-Net[2] [13, 14] was considered as the maximum complexity of the neural network architecture (Figure 11).

In this case, a 240 x 432 pixel image dataset made it possible to work with batches 4, 8 and 16. The maximum accuracy with batch 16 and a learning step of 0.001 was 86% at the 18th epoch, and

with batch 8 it was 88.5% at the same epoch 18. Thus, the U-Net^2 neural network demonstrated more intensive learning.

Table 1
The resulting accuracy of U-Net++ for different batch sizes

Batch	Accuracy	Epoch
8	85.2	33
16	71.5	18
32	83.8	76
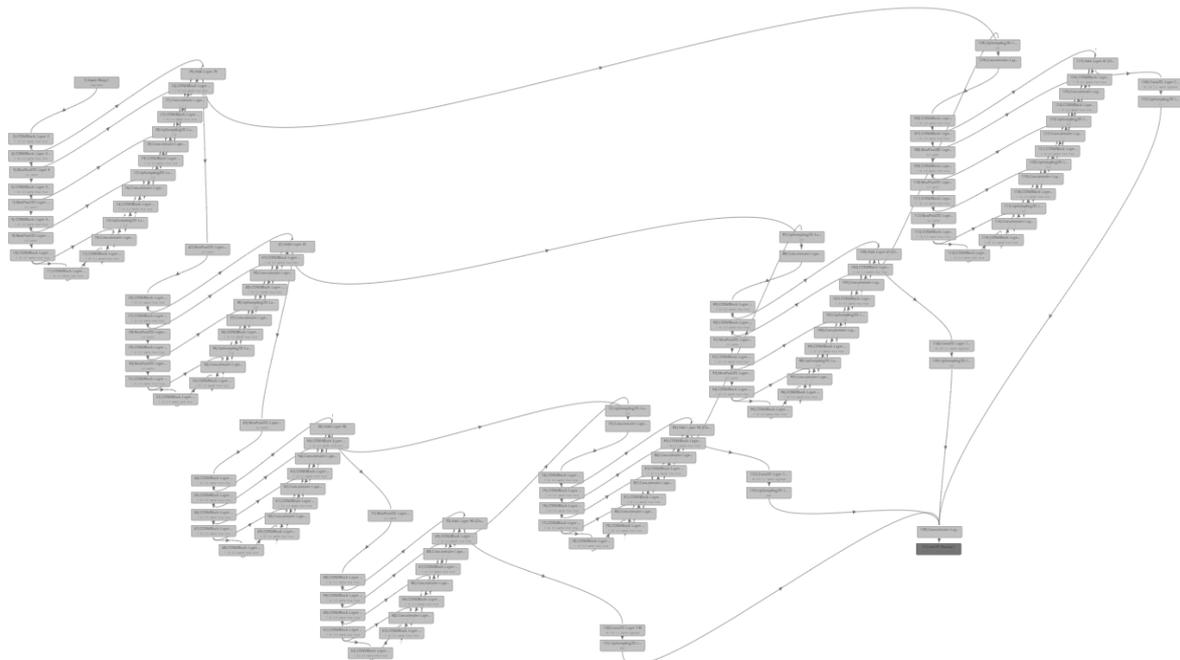
Figure 11: A general view on the architecture of "U-Net^2" from the Terra AI framework.

Since in batch 32, during learning with 240 x 432 images, an insufficient resource error occurs, with this batch, a transition was made to a smaller image format of 128 x 224 pixels. At the same time, an accuracy of 85.8% was obtained at the 71st epoch.

A comparison of the architectures of all considered neural networks is presented in Table. 2. As you can see, a larger architecture does not necessarily give a better result.

Table 2
The comparison of used neural networks

Architecture	Total parameters	Trainable parameters	Non-trainable parameters
PSPBlock2D	34,429,058	34,426,498	2,560
U-Net++	2,084,370	2,081,042	3,328
U-Net^2	682,290	678,706	3,584
PSP on Figure 1	923,266	923,266	0
PSP on Figure 2	574,158	574,152	6
Large U-Net	31,060,226	31,046,530	13,696
Medium U-Net	1,948,226	1,944,802	3,424
Light U-Net	1,869,826	1,866,882	2,944

The hardware implementation can be based on the Raspberry Pi Zero processor board, the neural network on the ESP32, and other solutions (Figure 12) proposed in [15]. Alternatively, you can use the ESP32-CAM module (Figure 13) [16], which is currently the most cost-effective option for implementing edge IoT (Figure 14).
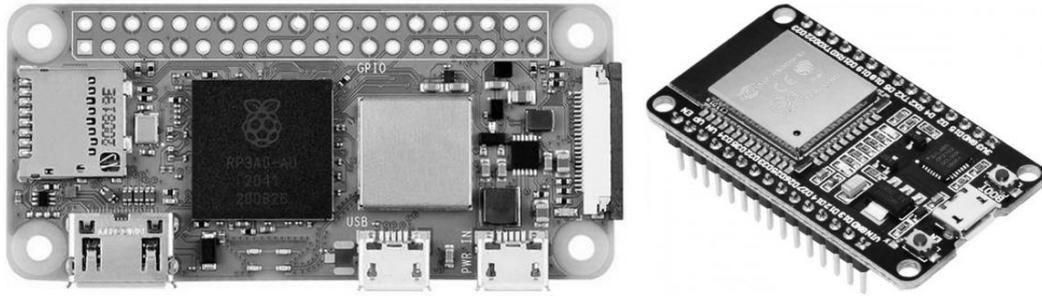
Figure 12: Raspberry Pi Zero 2W and ESP32

Figure 13: ESP32-CAM

Figure 14: Options for mounting it on meters

4. Perspectives sing of Further Research

Given the relevance of edge computing, it is advisable to explore the possibility of implementing segmentation and recognition based on TensorFlow Lite on devices such as ESP32-CAM.

The approach considered in the paper can be used not only in the interest of creating digital infrastructure. A promising direction is the development of uncrewed platforms that use vehicles initially oriented towards human control. At the same time, to read the indicators of their sensors, for example, a speedometer, engine speed, oil pressure, etc. video cameras with neural networks can be used, similar to the options discussed here for household meters.

The authors plan to continue further research on the possibility of using neural networks based on the use of Object Detection technology with marking digital displays using Bounding Boxes, using the results, for example, [17]. In addition, it is also of interest to generalize the approach considered here to the case of pointer analog devices and the use of pre-trained neural networks for image classification in the structure of a neural network.

5. Conclusion

The presence of outdated energy accounting equipment in the infrastructure makes it impossible to fully realize integration with the IoT ecosystem. Consequently, the transition to Industry 4.0can be very complicated and choosing the right solution path at the design stage will play a key role in the future.

The use of optical recognition of analog meter readings ensures minimal interference in the existing production process, and most importantly, without stopping it or stopping its monitoring. Therefore, such solutions are quite popular.

To eliminate edge restrictions computing data processing model in the IoT ecosystem should be based on fog computing. In this case, it becomes possible to perform an image segmentation procedure before recognition, including based on neural networks, the architectures of which are very complex for edge computing.

The paper considers an approach based on the modification of PSP or U-Net and U-Net^2. To evaluate the synthesized architectures, the value of the accuracy index on the validation set was used. Its maximum value is 88.8% when using a lightweight U-Net neural network and a learning image format of 224x128 pixels.

The proposed solutions can be used for other AI + IoT applications.

6. References

[1]   Gaz-counter. URL: https://github.com/maleficxp/gaz-counter.
[2]   AI-on-the-edge-device. URL: https://github.com/jomjol/AI-on-the-edge-device.
[3]   Analog meters in the digital enterprise: change or integrate? URL: https://habr.com/ru/company/lanit/blog/676240/.
[4]   F. Yang, Q. Sun, H. Jin and Z. Zhou, Superpixel segmentation with fully convolutional networks, in: Proceedings of the IEEE Conf. on Computer Vision and Pattern Recognition, 2020, pp. 13964-13973.
[5]   V. Slyusar, M. Protsenko, A. Chernukha, V. Melkin, O. Petrova, M. Kravtsov, S. Velma, N. Kosenko, O. Sydorenko and M. Sobol, Improving a neural network model for semantic segmentation of images of monitored objects in aerial photographs, Eastern-European Journal of Enterprise Technologies, vol. 2, no. 6 (114), 2021, pp. 86-95. doi:10.15587/1729-4061.2021.248390.
[6]   R. Kucev, Water Meters Dataset. Hot and cold water meters dataset. URL: https://www.kaggle.com/datasets/tapakah68/yandextoloka-water-meters-dataset.
[7]   H. Zhao, J. Shi, X. Qi, X. Wang and J. Jia, Pyramid Scene Parsing Network. URL: https://arxiv.org/abs/1612.01105.

[8] O. Ronneberger, P. Fischer, and T. Brox, U-Net: Convolutional Networks for Biomedical Image Segmentation. URL: https://arxiv.org/pdf/1505.04597.pdf.

[9] W. Jwaid, Z. Al-Husseini and A. Sabry, Development of brain tumor segmentation of magnetic resonance imaging (MRI) using U-Net deep learning, Eastern-European Journal of Enterprise Technologies, vol. 4, no. 9 (112), 2021, pp. 23-31. doi:10.15587/1729-4061.2021.238957.

[10] N. Singh and K. Nongmeikapam, Semantic segmentation of satellite images using deep-UNet, Arabian Journal for Science and Engineering, 2022, pp. 1-13.

[11] A. Soni, R. Koner, and V. Villuri, M-Unet: Modified U-Net segmentation framework with satellite imagery, in: Proceedings of the Global AI Congress 2019, Springer, 2020, pp. 47-59.

[12] E. Irwansyah, Y. Heryadi, and A. Gunawan, Semantic image segmentation for building detection in urban area with aerial photograph image using U-Net models, in: Proceedings of the 2020 IEEE Asia-Pacific Conf. on Geoscience, Electronics and Remote Sensing Technology (AGERS), 2020, pp. 48-51.

[13] X. Qin,; Z. Zhang, C. Huang, M. Dehgan, O. Zaiane and M. Jagersand, U^2-Net: Going deeper with nested U-structure for salient object detection. pattern recognition. 2020, 106, 107404.

[14] F. Ge, G. Wang, G. He, D. Zhou, R. Yin and L. Tong, A Hierarchical Information Extraction Method for Large-Scale Centralized Photovoltaic Power Plants Based on Multi-Source Remote Sensing Images. URL: https://www.mdpi.com/2072-4292/14/17/4211.

[15] H. Padmasiri, J. Shashirangana, D. Meedeniya, O. Rana and C. Perera, Automated License Plate Recognition for Resource-Constrained Environments. URL: https://www.mdpi.com/1424-8220/22/4/1434/htm.

[16] ESP32-CAM. URL: https://www.espressif.com/en/news/ESP32_CAM.

[17] V. Slyusar, M. Protsenko, A. Chernukha, S. Gornostal, S. Rudakov, S. Shevchenko, O. Chernikov, N. Kolpachenko, V. Timofeyev and R. Artiukh, Construction of an advanced method for recognizing monitored objects by a convolutional neural network using a discrete wavelet transform, Eastern-European Journal of Enterprise Technologies, vol. 4, no. 9 (112), 2021, pp. 65-77. doi:10.15587/1729-4061.2021.238601.