**Вісник Тернопільського національного технічного університету**
**https://doi.org/10.33108/visnyk_tntu**
Scientific Journal of the Ternopil National Technical University
*2025, № 4 (120) https://doi.org/10.33108/visnyk_tntu2025.04*
*ISSN 2522-4433. Web:* ***visnyk.tntu.edu.ua***

**UDC 004.932.2:004.93'1**


# OPTIMIZATION OF THE ARCHITECTURE AND HYPERPARAMETERS OF THE U-NET MODEL TO IMPROVE THE QUALITY OF BIOLOGICAL OBJECTS SEGMENTATION

**Anton Kovalenko; Valerii Severyn**

*National Technical University «Kharkiv Polytechnic Institute», Kharkiv, Ukraine*

***Abstract.*** *The paper addresses the optimization of the U-Net architecture and hyperparameters to improve the segmentation accuracy of biological objects in microscopic images. Image segmentation is formalized as the optimization of a parametric mapping that minimizes a compound loss function combining binary cross-entropy and the Dice coefficient, ensuring better boundary detection and robustness to class imbalance. A modified U-Net with three encoding and decoding levels was developed to balance computational efficiency and segmentation quality. During preprocessing, segmentation masks were refined by filling object contours, which improved model stability and mask accuracy. Experiments on microscopic cell images showed that the combined loss function achieved a mean Dice index of 0.9037, outperforming binary cross-entropy alone. The Adam optimizer provided better convergence and stability than RMSProp, confirming its effectiveness for small datasets. The proposed approach can be applied to automated cell analysis and biomedical diagnostic systems.*

***Key words:*** *mathematical model, artificial neural network, object detection, computer vision, information technology.*

*Submitted 07.09.2025*
*Revised 21.11.2025*
*https://doi.org/10.33108/visnyk_tntu2025.04.021* *Published 27.01.2026*

## 1. INTRODUCTION

Identifying objects in microscopic images is an extremely important part of many fields of science and technology, including medicine, biology, chemistry, materials science, micro- and nanoelectronics technology, geology, cosmology and others.

Localization of bio-objects in microscopic images is the process of detecting and determining the location of objects such as cells, proteins, or structures in images obtained using microscopes. This concept is important for many fields, where accurate object localization can be crucial for data analysis and diagnostics. The localization [1] of biological objects can be performed using various image processing methods, such as segmentation, edge detection, artificial intelligence, or neural networks [2].

Segmentation is one of the tasks of computer vision that has found wide application in medical diagnostics, cell biology, and biotechnology. Segmentation of biological objects in microscopic images allows for the automation of cell structure analysis, quantitative measurements, and the detection of pathological changes. Classic image processing methods [3] (threshold binarization, filtering, or morphological operations) are limited in their ability to correctly process images with complex textures, uneven lighting, or overlapping objects.

The use of convolutional neural networks [4, 5, 6], in particular the U-Net architecture, has opened up new possibilities for high-precision segmentation algorithms that can work even on small datasets thanks to the use of a symmetric encoder-decoder structure and skip connections. This study investigates the effectiveness of the U-Net model with modified

encoder depth and a combined loss function, and evaluates the impact of different optimizers on the quality of cell structure segmentation.

## 2. EXPERIMENTAL METHODS

The segmentation problem can be defined as follows. Let the input image be given:

$$I : \Omega \subset \mathbb{R}^2 \to \mathbb{R}^C,$$

where $\Omega$ is the pixel coordinate space, and $C$ is the number of channels.

It is necessary to construct a segmentation mask

$$M : \Omega \to \{0,1,...,K-1\},$$

where $K$ is the number of classes (for example, $K = 2$ for binary segmentation «object/background»).

The task is to approximate the mapping

$$f_\theta : \mathbb{R}^{H \times W \times C} \to \{0,1,...,K-1\}^{H \times W},$$

parameterized by a set of parameters $\theta$, where $\hat{M} = f_\theta(I)$ is a predicted mask of objects in the image. In other words $f_\theta$ is an artificial neural network model that accepts images $I$ of size $H \times W$ with $C$ color channels and outputs a class map (segmentation mask) of size $H \times W$, in which each pixel belongs to one of $K$ classes. In turn, $\theta$ presents the model parameters (for a neural network, these are the weights and biases of all layers) that the network «trains» to approximate the mapping $I \to M$, and the training process itself involves selecting such $\theta$, that $f_\theta(I)$ is as close as possible to the true mask $M$. In turn, $\hat{M}$ is a predicted mask for provided image $I$, calculated using the parameters $\theta$.

A set of labeled data is used to train the model:

$$D = \{(I_i, M_i)\}_{i=1}^N,$$

where each element contains an input image $I_i$ and its ground truth mask $M_i$.

The learning process consists of minimizing the loss function

$$Loss(\theta) = \frac{1}{N} \sum_{i=1}^{N} l(f_\theta(I_i), M_i),$$

which measures the difference between the predicted and actual masks.

Since the mask is a binary image, it is logical to use the binary cross-entropy function as the loss function:

$$Loss_{BCE} = -\frac{1}{|\Omega|} \sum_{(x,y)\in\Omega} [M(x,y) \log \hat{M}(x,y) + (1 - M(x,y)) \log(1 - \hat{M}(x,y))]. \quad (1)$$

Thus, the model parameters are defined as:

$$\theta^* = \arg\min_\theta Loss(\theta).$$

The process of constructing a mask for an object includes the following main stages:

1. Preliminary image processing such as normalization, noise filtering, contrast enhancement.

2. Feature extraction or application of a convolutional neural network $f_\theta(I)$.

3. Probability map calculation:

$$P(x,y) = \Pr(M(x,y) = 1 \mid I).$$

4. Thresholding (binarization):

$$\hat{M}(x,y) = \begin{cases} 1, \text{ if } P(x,y) \geq t, \\ 0, \text{ else.} \end{cases}$$

5. Post-processing of results (optional) – morphological operations, watershed or region-growing to refine boundaries.

The quality of the constructed mask is evaluated based on the similarity metrics between the predicted mask $\hat{M}$ and ground truth one $M$ determined using the Dice coefficient [7]:

$$Dice = \frac{2\left|M \cap \hat{M}\right|}{\left|M\right| + \left|\hat{M}\right|} \tag{2}$$

or IoU [8] (Intersection over Union) metrics

$$IoU = \frac{\left|M \cap \hat{M}\right|}{\left|M \cup \hat{M}\right|},$$

where $M \cap \hat{M}$ is the area of overlap between the ground truth mask and the predicted mask, i.e., the pixels where the model «guessed» the object correctly; $M \cup \hat{M}$ is the combined area of the ground truth and predicted masks.

There is an exact relationship between the Dice coefficient and IoU metrics:

$$Dice = \frac{2 \cdot IoU}{IoU + 1}$$

and vice versa

$$IoU = \frac{Dice}{2 - Dice}.$$

That is, these two metrics evaluate the same overlap, but the Dice index gives higher values for the same overlap. In this work, the Dice index will be used to evaluate the quality of the predicted mask.

The classic architecture for segmenting biological objects (Fig. 1), based on the use of artificial neural networks in general and CNN in particular, is the U-Net network [9]. The general structure of the network consists of two main parts [10]: the left side is responsible for feature extraction (the so-called encoder, or dimension reduction stage), while the right side is responsible for spatial data recovery (decoder, or dimension enhancement stage).

In the encoder part, the model obtains the semantic characteristics of the image using sequential convolution and pooling operations.

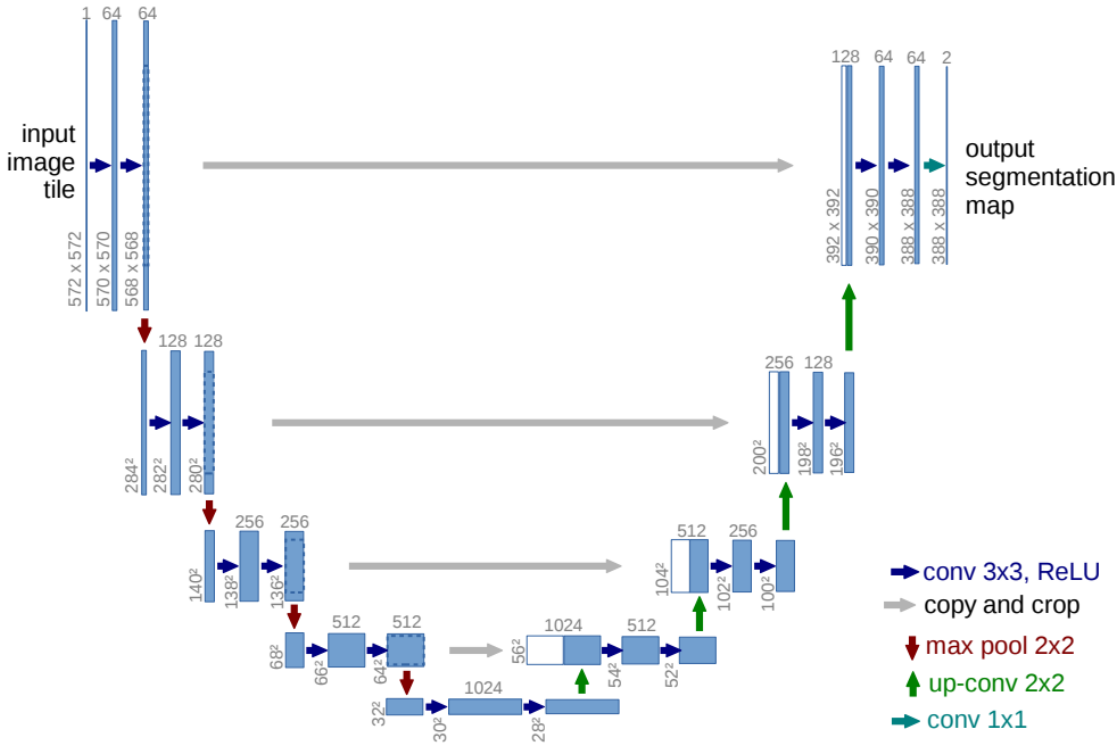



**Figure 1.** Classical U-Net architecture [9]

At each stage, the image undergoes two convolution operations, after which the number of channels increases by 64, and then maximum pooling is performed, which reduces the spatial dimensions (length and width) by half.

In the classic version [9], after four consecutive stages of double convolution and max pooling, two convolution operations with a 3x3 kernel are applied to the image to obtain the final feature map – the so-called U-Net bottleneck.

At the decoding stage, or dimensionality enhancement, processing begins at the deepest level of the network. After each deconvolution operation (UpSampling2D 2×2), the resulting feature map is combined with the corresponding feature map from the encoder part of the same level. This allows high- and low-level information to be combined. Next, two consecutive convolutions with a 3×3 kernel are performed, after which one upsampling step is completed. This architecture creates a mechanism for combining features from different levels, allowing the network to enrich spatial information and improve segmentation accuracy.

However, in a series of works [11, 12], there are experiments with a «shallow» version of the architecture, where 3 levels of downsampling are used in the encoder instead of 4.

In this study, to detect objects using binary classification (i.e., «objects/background»), three levels of downsampling were applied in the encoder and, accordingly, 3 levels of upsampling in the decoder. The Python programming language was used to implement the architecture, with the corresponding libraries (numpy, opencv, scikit-learn, tensorflow, matplotlib).

Each level of the encoder (Fig. 2) consists of 2 layers of Conv2D convolution with the ReLU activation function and one layer of MaxPooling2D max pooling. We will also use the padding='same' parameter, which adds a «frame» of zeros around the edges of the image so that the spatial dimension is not lost after convolution. This is very convenient in U-Net so that the skip connections between the encoder and decoder match in size.

```
# ENCODER
c1 = layers.Conv2D(16, 3, activation='relu', padding='same')(inputs)
c1 = layers.Conv2D(16, 3, activation='relu', padding='same')(c1)
p1 = layers.MaxPooling2D((2,2))(c1)

c2 = layers.Conv2D(32, 3, activation='relu', padding='same')(p1)
c2 = layers.Conv2D(32, 3, activation='relu', padding='same')(c2)
p2 = layers.MaxPooling2D((2,2))(c2)

c3 = layers.Conv2D(64, 3, activation='relu', padding='same')(p2)
c3 = layers.Conv2D(64, 3, activation='relu', padding='same')(c3)
p3 = layers.MaxPooling2D((2,2))(c3)
```

**Figure 2.** Encoder layers

In turn, the U-Net decoder (Fig. 3), which performs sequential spatial dimension elevation and concatenating with the corresponding encoder layers (skip connections), consists of upsampling layers, which increase the height and width of the tensor by 2 times; Concatenate, which combines (by channels) the raised tensor with the corresponding encoder layer to transfer details from earlier layers (skip connection), and two Conv2D convolutional layers with ReLU activation function to process the combined tensor and extract features after upsampling.

```
# DECODER
u5 = layers.UpSampling2D((2,2))(c4)
u5 = layers.Concatenate()([u5, c3])
c5 = layers.Conv2D(64, 3, activation='relu', padding='same')(u5)
c5 = layers.Conv2D(64, 3, activation='relu', padding='same')(c5)

u6 = layers.UpSampling2D((2,2))(c5)
u6 = layers.Concatenate()([u6, c2])
c6 = layers.Conv2D(32, 3, activation='relu', padding='same')(u6)
c6 = layers.Conv2D(32, 3, activation='relu', padding='same')(c6)

u7 = layers.UpSampling2D((2,2))(c6)
u7 = layers.Concatenate()([u7, c1])
c7 = layers.Conv2D(16, 3, activation='relu', padding='same')(u7)
c7 = layers.Conv2D(16, 3, activation='relu', padding='same')(c7)
```

**Figure 3.** Decoder of the U-Net architecture

The deepest layer between the encoder and the decoder (Fig. 4) is so-called bottleneck – the central «compression» of network information. This is where the neural network «sees» the entire context of the image in a reduced size. It is represented by two Conv2D layers that help the network identify multi-channel features before beginning to restore the spatial dimension in the decoder.

```
# BOTTOM
c4 = layers.Conv2D(128, 3, activation='relu', padding='same')(p3)
c4 = layers.Conv2D(128, 3, activation='relu', padding='same')(c4)
```

**Figure 4.** Central block of architecture

As the output layer (Fig. 5), we also use a Conv2D convolution with a 1×1 kernel, which reduces the number of channels to 1, since we have binary segmentation with a sigmoid activation function that converts pixel values to the range [0, 1] to obtain the probability of belonging to the target class. The output of the outputs layer is the final prediction mask, ready for comparison with the ground truth.

```
outputs = layers.Conv2D(1, 1, activation='sigmoid')(c7)
```

**Figure 5.** The output layer of architecture

Several experiments were conducted with a dataset for binary classification [13]. This image set consists of 60 images of CHO (Chinese Hamster Ovary) cells obtained using an Olympus Cell-R microscope with a 20x lens. In accordance with the study [14], the dataset was divided into training, validation, and testing sets in a ratio of 80% – 10% – 10%.

The reference masks in the set are manually segmented images in which the cells are highlighted with contours. As a preliminary processing of the dataset, the contours of objects on the reference masks were filled in. The filling was performed using the algorithm in Fig. 6.

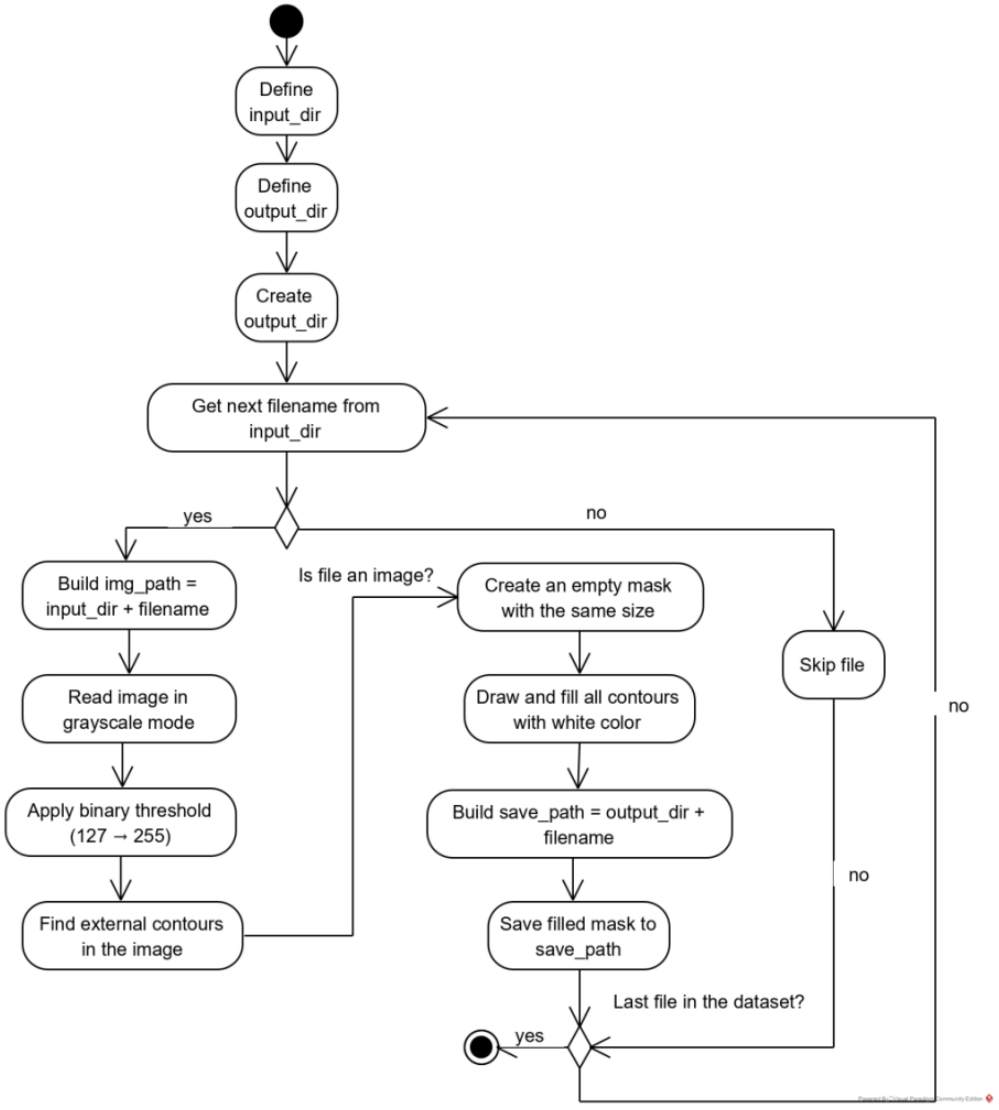



**Figure 6.** Algorithm for filling objects in images of ground truth masks

The initial appearance of objects on masks and the result of preliminary processing for filling circles are shown in Fig. 7.

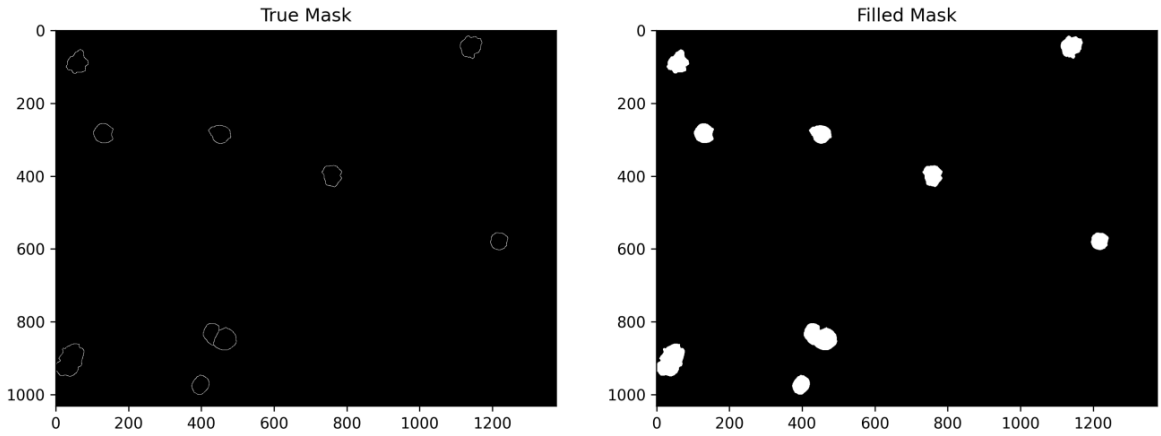



**Figure 7.** Filled ground truth masks used for training

Several experiments were conducted with different hyperparameters: RMSProp and Adam optimizers were used with a learning rate of 0.01, a batch size of 8 [15], and a loss function: binary cross-entropy and a combined function

$$Loss = \alpha \cdot BCE + \beta \cdot (1 - Dice),$$

where $BCE$ is the binary cross entropy (1), $Dice$ is the Dice function (2), $\alpha$ and $\beta$ are weights for balancing the contribution of each component.

## 3. RESULTS AND DISCUSSION

To evaluate the impact of architectural complexity on segmentation quality and computational efficiency, a comparative experiment was conducted using two configurations of the U-Net model: the classical four-level U-Net [9], and an optimized three-level version with a reduced number of convolutional blocks and parameters. Both models were trained under identical experimental conditions on a computer with the parameters presented in Table 1.

**Table 1**

Hardware Specifications Used for U-Net Training.

| Parameter | Value |
|---|---|
| RAM | 16 GB |
| Speed | 3200MHz |
| Processor | 12th Gen IntelI CoreI i5-12599H |
| Freq | 2.5GHz |
| Operating System | Windows 10 |

The models were trained for 50 epochs on 256×256 images. The results are presented in Table 2, where $\text{Dice}_{tr}$, $\text{Dice}_{val}$ are the Dice indices of training and validation on the last epoch, $\text{Dice}_{mean}$ is the mean Dice index on the validation set:

$$Dice_{mean} = \frac{1}{N}\sum_{i=1}^{N} Dice_i .$$

It should be noted that when predicting masks, accuracy refers to pixel accuracy, i.e., the proportion of pixels that have the same value in the prediction and in the reference mask.

**Table 2**

Models training results.

| Architecture | | Loss function | |
|---|---|---|---|
| | | Binary cross-entropy | Binary cross-entropy + Dice |
| Classical 4-level U-Net | Number of parameters | 31,378,945 (119.70 MB) | |
| | CPU usage | ≈69% | ≈69% |
| | Memory usage | ≈95% | ≈95% |
| | Training time | 1h 35m 45s | 1h 37m 15s |
| | $Dice_{tr}$ | 0.8786 | 0.9240 |
| | $Dice_{val}$ | 0.8586 | 0.9128 |
| | $Dice_{mean}$ | 0.8532 | 0.9098 |
| Optimized 3-level U-Net | Number of parameters | 1,461,893 (5.58 MB) | |
| | CPU usage | ≈55% | ≈55% |
| | Memory usage | ≈60% | ≈60% |
| | Training time | 3m 42c | 3m 54c |
| | $Dice_{tr}$ | 0.8798 | 0.9217 |
| | $Dice_{val}$ | 0.8717 | 0.9099 |
| | $Dice_{mean}$ | 0.8702 | 0.9037 |

The classical U-Net contains 31,378,945 trainable parameters (119.70 MB), but the optimized architecture includes only 1,461,893 parameters (5.58 MB). This corresponds to a 21-fold reduction in model size, which significantly influences computational load. During training, the classical architecture utilized approximately 69% of CPU resources and 95% of RAM, almost approaching the hardware limits of the system. In contrast, the optimized network required only 55% of CPU load and 60% of RAM, indicating substantially lower resource consumption and greater system stability.

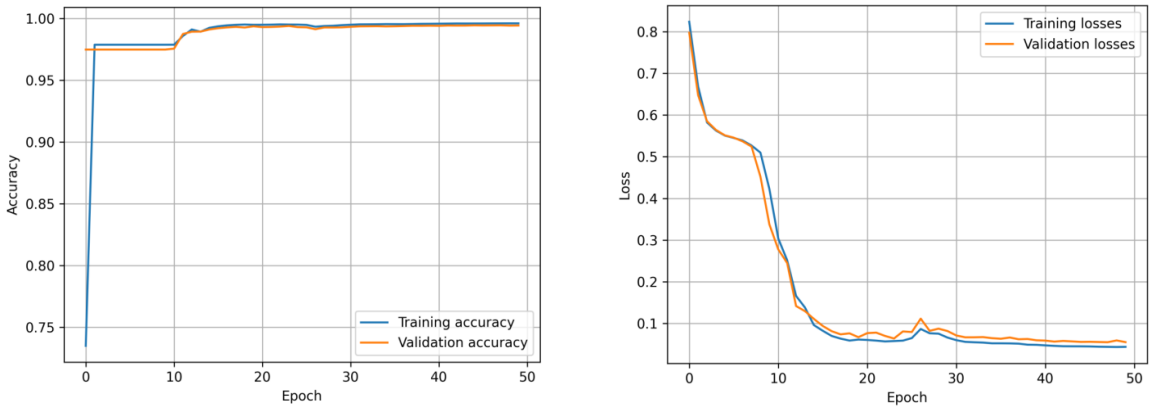
Training time further demonstrates the advantage of the lightweight architecture. The classical model required more than 1.5 hour for both binary cross-entropy and combined loss functions. Meanwhile, the optimized model completed training in merely 3 min 42 s and 3 min 54 s, respectively. Thus, the optimized U-Net trains approximately 25 times faster.

Despite the significant difference in computational complexity, both architectures achieved comparable segmentation accuracy. For the classical U-Net, the mean Dice coefficient reached 0.9098 when using the combined loss function, while the optimized architecture achieved 0.9037, reflecting only a 0.6% decrease in accuracy. A similar trend is observed for the training and validation metrics, where the optimized network slightly lags behind the classical one, but the performance gap remains minimal and not critical for practical applications.

The results indicate that although the classical U-Net demonstrates marginally higher segmentation quality, the computational expense associated with its training is disproportionately large relative to the obtained improvement. The optimized architecture offers a substantially more favorable balance between accuracy, speed, and hardware

requirements. The reduced number of parameters allows the model to train efficiently on standard personal computers without specialized GPUs, accelerates experimentation with hyperparameters, and decreases the risk of memory saturation.

The training and validation accuracy and loss curves for the optimized 3-level U-Net architecture are given in Fig. 8:




a) b)

**Figure 8.** Accuracy a) and losses b) of network training

The curve of changes in the Dice coefficient is shown in Fig. 9.

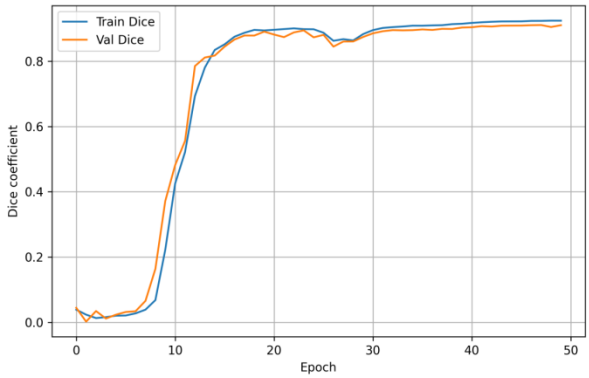



**Figure 9.** Changes in the Dice index during network training

After training the network, we have the predicted mask (Fig. 10)

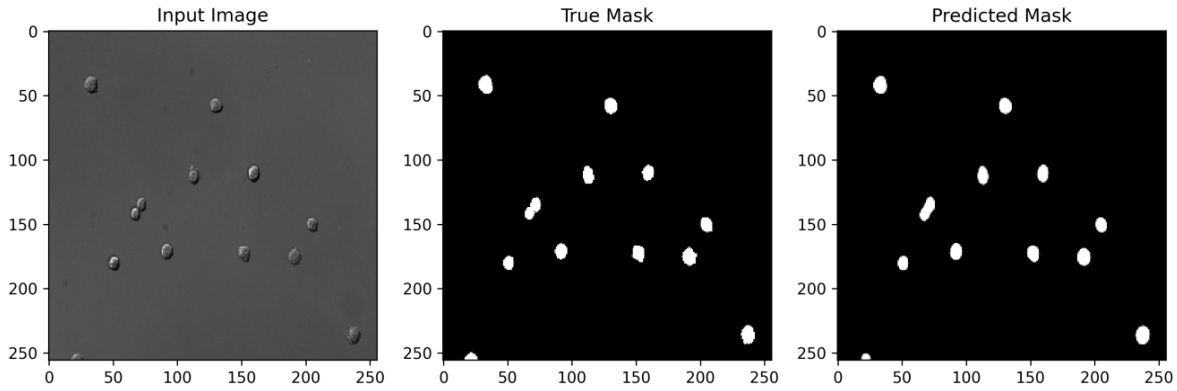



**Figure 10.** Original image, ground truth mask, and predicted mask

## 4. CONCLUSIONS

The main results of the study are as follows:

1. The image segmentation process was formalized as an optimization task of learning a parametric mapping $f : I \to M$ that minimizes a composite loss function combining binary cross-entropy and Dice coefficient, ensuring increased boundary detection accuracy and robustness to class imbalance. The proposed formal model defines a structured segmentation pipeline and establishes analytical relationships between evaluation metrics, allowing for a consistent quantitative comparison of different network configurations.

2. A modified U-Net architecture with three levels of downsampling and upsampling was developed and trained, providing an optimal balance between computational complexity and the quality of biological object segmentation.

3. The comparative analysis demonstrates that the optimized U-Net architecture provides segmentation accuracy comparable to that of the classical model, while requiring drastically fewer computational resources and achieving training times shorter by more than an order of magnitude. Considering the minimal loss in performance and significant gains in efficiency, the optimized architecture is more suitable for biomedical image segmentation tasks under hardware limitations and for scenarios that require rapid model iteration and deployment.

4. Experimental results showed that the use of a combined loss function (Binary Cross-entropy + Dice Loss) allows achieving the highest average Dice index (0.9037) compared to the variant where only binary cross-entropy is used.

5. Scientific novelty of the work lies in the use of a simplified U-Net architecture, which maintains high segmentation accuracy, as well as in the implementation of a combined loss function, which improves the quality of cell contour reproduction and allows for more accurate identification of object boundaries.

6. Further development of the research involves the application of post-processing algorithms to improve the separation of stuck cells, as well as the integration of attention modules and transformer architectures to increase the generalization ability of segmentation models and improve accuracy in complex biomedical scenes.

**References**


1. Kovalenko S. M., Kutsenko O. S., Kovalenko S. V., Kovalenko A. S. (2024) Approach to the Automatic Creation of an Annotated Dataset for the Detection, Localization and Classification of Blood Cells in an Image. Radio Electronics, Computer Science, Control, (1), 128. https://doi.org/10.15588/1607-3274-2024-1-12
2. Kovalenko S., Kovalenko S., Kutsenko A., Godlevskyi M., Severin V. (2024). Kovalenko A. Methodology for Creating Annotated Datasets of Biological Objects in Microscopic Images. 2024 IEEE 5th KhPI Week on Advanced Technology (KhPIWeek), Kharkiv, Ukraine, pp. 1–6. https://doi.org/10.1109/KhPIWeek61434.2024.10878016
3. Kovalenko S., Kovalenko S., Mikhnova O., Kovalenko A., Pelikh D., Severin V. (2023). An Approach to Blood Cell Classification Based on Object Segmentation and Machine Learning. 2023 IEEE 4th KhPI Week on Advanced Technology (KhPIWeek), Kharkiv, Ukraine, pp. 1–6. https://doi.org/10.1109/KhPIWeek61412.2023.10312903
4. Jin S., Yu S., Peng J. et al. A novel medical image segmentation approach by using multi-branch segmentation network based on local and global information synchronous learning. Sci Rep 13, 6762. 2023. https://doi.org/10.1038/s41598-023-33357-y
5. Totosko O., Stukhliak D., Stukhliak P. (2025) Usage of neural networks for analysis and processing of experimental research of composite materials. Scientific Journal of TNTU (Tern.), vol. 118, no. 2, pp. 42–55. https://doi.org/10.33108/visnyk_tntu2025.02.042
6. Stefanyshyn V., Stefanyshyn I., Pastukh O., Kulikov S. (2024) Comparison of the accuracy of machine learning algorithms for brain-computer interaction based on high-performance computing technologies. Scientific Journal of TNTU (Tern.), vol. 115, no. 3, pp. 82–90. https://doi.org/10.33108/visnyk_tntu2024.03.082
7 Jha S., Son L. H., Kumar R., Priyadarshini I., Smarandache F., Long H. V., (2019) Neutrosophic image segmentation with Dice Coefficients. Measurement, vol. 134, pp. 762–772, ISSN 0263-2241. https://doi.org/10.1016/j.measurement.2018.11.006

8. Müller D., Soto-Rey I., Kramer F. (2022) Towards a guideline for evaluation metrics in medical image segmentation. BMC Res Notes. 15. 210. https://doi.org/10.1186/s13104-022-06096-y
9. Ronneberger O., Fischer P., Brox T. U-Net: Convolutional Networks for Biomedical Image Segmentation. In: Navab, N., Hornegger, J., Wells, W., Frangi, A. (eds) Medical Image Computing and Computer-Assisted Intervention – MICCAI 2015. MICCAI 2015. Lecture Notes in Computer Science, vol. 9351. 2015. Springer, Cham. https://doi.org/10.1007/978-3-319-24574-4_28
10. Wang Y., Kong J., Zhang H. U-Net: A Smart Application with Multidimensional Attention Network for Remote Sensing Images. Scientific Programming 2022.1 2022. 1603273. https://doi.org/10.1155/2022/1603273
11 Jena B., Jain S., Nayak G. K., Saxena S. (2023) Analysis of depth variation of U-NET architecture for brain tumor segmentation. Multimedia Tools Appl. 82, 7, pp. 10723–10743. https://doi.org/10.1007/s11042-022-13730-1
12 Gkologkinas G. D., Ntouros K., Protopapadakis E., Rallis I. (2025) A Comparative Analysis of U-Net Architectures with Dimensionality Reduction for Agricultural Crop Classification Using Hyperspectral Data. Algorithms, 18, 588. https://doi.org/10.3390/a18090588
13 Chinese Hamster Ovary Cells dataset. https://bbbc.broadinstitute.org/BBBC030.
14. Furtado P. (2021) Testing Segmentation Popular Loss and Variations in Three Multiclass Medical Imaging Problems. J Imaging, 27;7(2):16. PMID: 34460615; PMCID: PMC8321275. https://doi.org/10.3390/jimaging7020016
15 Kartowisastro I. H., Latupapua J. (2023) A comparison of Adaptive Moment Estimation (Adam) and RMSProp optimisation techniques for wildlife animal classification using convolutional neural networks // Revue d'Intelligence Artificielle, vol. 37, no. 4, pp. 1023–1030. https://doi.org/10.18280/ria.370424

**УДК 004.932.2:004.93'1**

# ОПТИМІЗАЦІЯ АРХІТЕКТУРИ ТА ГІПЕРПАРАМЕТРІВ МОДЕЛІ U-NET ДЛЯ ПОКРАЩЕННЯ ЯКОСТІ СЕГМЕНТАЦІЇ БІОЛОГІЧНИХ ОБ'ЄКТІВ

**Антон Коваленко; Валерій Северин**

*Національний технічний університет «Харківський політехнічний інститут», Харків, Україна*

***Резюме.*** *Розглянуто задачу оптимізації архітектури та гіперпараметрів моделі U-Net для підвищення точності сегментації біологічних об'єктів на мікроскопічних зображеннях. Процес сегментації зображень розглядається як задача оптимізації параметричного відображення* $f : I \to M$ *, що мінімізує складену функцію втрат, яка поєднує бінарну крос-ентропію та коефіцієнт Дайса. Такий підхід забезпечує підвищену точність визначення меж об'єктів та стійкість до дисбалансу класів. Розроблено модифіковану архітектуру U-Net з трьома рівнями кодування та декодування, що дозволяє зменшити глибину мережі без втрати точності. Такий підхід забезпечує оптимальне співвідношення між обчислювальною ефективністю, швидкістю навчання та якістю сегментації. Експериментальні дослідження проводилися на наборі мікроскопічних зображень клітин, що містять контури біологічних структур. У процесі попереднього опрацювання даних виконано налаштування масок сегментації шляхом заповнення контурів об'єктів, що дало змогу сформувати повні цільові області замість лише контурних меж. Це суттєво підвищило стабільність навчання моделі та точність прогнозування піксельних масок. Отримані результати показали, що використання комбінованої функції втрат дозволяє отримати кращі метрики якості при навчанні моделі й середнього індексу Дайса 0,9037 на валідаційному наборі, що перевищує результати, отримані при використанні лише бінарної перехресної ентропії. Оптимізатор Adam забезпечив кращу збіжність та стабільність результатів порівняно з RMSProp, підтверджуючи його ефективність для сегментації на обмежених наборах даних. Запропонований підхід дозволяє зменшити глибину моделі без втрати точності сегментації й може бути використаний для автоматизованого аналізу клітинних структур, оцінювання морфологічних змін і розроблення систем комп'ютерної діагностики у біомедичних дослідженнях.*

***Ключові слова:*** *математична модель, штучна нейронна мережа, виявлення об'єктів, комп'ютерний зір, інформаційна технологія.*