

# Improving the efficiency of damaged buildings detection based on ASPP technologies

Valerii Dymo<sup>1,\*†</sup>, Aleksandr Gozhyj<sup>1†</sup> and Irina Kalinina<sup>1†</sup>

<sup>1</sup> Petro Mohyla Black Sea National University, St. 68 Desanynykiv 10, Mykolaiv, 54000, Ukraine

## Abstract

The paper presents an increase in the efficiency of detecting damaged buildings on satellite images of the U-Net convolutional network model by modifying. Instead of the usual bottleneck, the use of atrous spatial pyramid pooling (ASPP) is proposed. As part of the study, the dataset was expanded to 100 images with dimensions of 512x512 pixels, and various augmentations were applied to increase the variability of the dataset, which contributed to more effective training on a limited dataset. Weighting coefficients for each image were also added to the dataset, which were used during training to solve the problem of the predominance of the number of pixels of one class over others. Models of different configurations with an ASPP layer were built and compared with the base U-Net model without ASPP. As a result of testing on the evaluation dataset, an increase in the mean IoU by 5.39% compared to the classical architecture was observed, as well as a significant reduction in overall losses and an increase in the mean IoU by about 2% on a separate testing dataset, which indicates a corresponding increase in the model's efficiency. The proposed architecture can be used in further studies of segmentation of images of buildings damaged by hostilities.

## Keywords

detection of damage buildings, semantic segmentation, U-Net, ASPP

## 1. Introduction

Recent studies [1-3] show that a large number of popular and effective neural network models for detecting damage due to various disasters are built using convolutional networks, among which YOLO, ResNet and U-Net stand out. At the same time, the application of such neural networks for complex detection or segmentation tasks, for example, as the detection of damaged or destroyed buildings for preliminary assessment of destruction, reveals existing difficulties: loss of accuracy in the case of a large number of objects of different sizes, insufficient recognition of object edges, etc.

Various approaches are used to improve efficiency. For example, the use of weighting coefficients helps to train the model with different numbers of images or pixels belonging to one of the classes. Another approach may be to increase the number of images through augmentation, or to change the parameter settings in the model itself – this may affect the model's ability to take into account more features from the images, but does not solve problems such as the model's inability to “focus” on specific regions of the image, or to extract information of different scalability, which requires the use of more complex image processing methods, changing the model architecture, etc.

The authors of the study [4] propose their own module Feature Pyramid Network, which can be used in various convolutional networks to extract image features for both object detection and

---

*MoMLeT-2025: 7th International Workshop on Modern Machine Learning Technologies, June, 14, 2025, Lviv-Shatsk, Ukraine*

<sup>\*</sup> Corresponding author.

<sup>†</sup> These authors contributed equally.

✉ dymovalery@gmail.com (V. Dymo); alex.gozhyj@gmail.com (A. Gozhyj); irina.kalinina1612@gmail.com (I. Kalinina)

ORCID 0009-0001-6024-3018 (V. Dymo); 0000-0002-3517-580X (A. Gozhyj); 0000-0001-8359-2045 (I. Kalinina)



© 2025 Copyright for this paper by its authors. Use permitted under Creative Commons License Attribution 4.0 International (CC BY 4.0).

segmentation. The module is built taking into account the separation of features “top-down”, which allows detecting both low-level features and high-level ones.

In studies [5-6], the use of spatial pyramid pooling was also proposed, which allows the use of datasets of different dimensions without losing the accuracy and efficiency of object detection in images. In turn, the authors of [6] propose atrous convolutions as a solution for expanding the receptive field by using larger gaps between kernel elements, which allows capturing remote features and more spatial information without increasing the number of parameters or reducing resolution. Combining the architecture of pyramid pooling, the authors proposed atrous spatial pyramid pooling (hereinafter ASPP is used), which allowed capturing more information of different scales by using parallel atrous convolutions with different rates.

The authors applied the created ASPP technique in their own model DeepLabv3+ [7], which extends the previous model by using an encoder-decoder architecture. This allowed to improve the segmentation of objects in images that have a complex structure and different scalability.

The authors of the following studies [8-10] applied the general principle of operation of ASPP to solve various tasks related to remote recognition of objects on the ground. Thus, the work [8] aims to build a horizontal U-shaped GAN model for segmentation of building images using the intermediate ASPP module, which allows to improve the localization and detection of buildings of different sizes.

In the study [9], a modified ASPP architecture is proposed using an additional feature extraction channel and changing the design of the dilation rate, as well as introducing an attention coordination mechanism. Despite the improvements obtained, the authors note the possibility of some limitations of the model regarding the influence of shadows on the accuracy of building and vegetation segmentation, which can be solved by more accurate color distribution, detection of object edges, etc.

In [10], a new Feature Residual Analysis Network model is presented, which offers a balance between sufficient accuracy and speed of feature extraction, using Feature Pyramid Pooling inspired by the corresponding ASPP module.

In turn, although the study [11] does not address the topic of building detection in images, the authors propose the application of U-Net using ASPP for segmentation of brain tumor images, and note the corresponding improvements in accuracy and reduction of losses relative to the basic U-Net architecture.

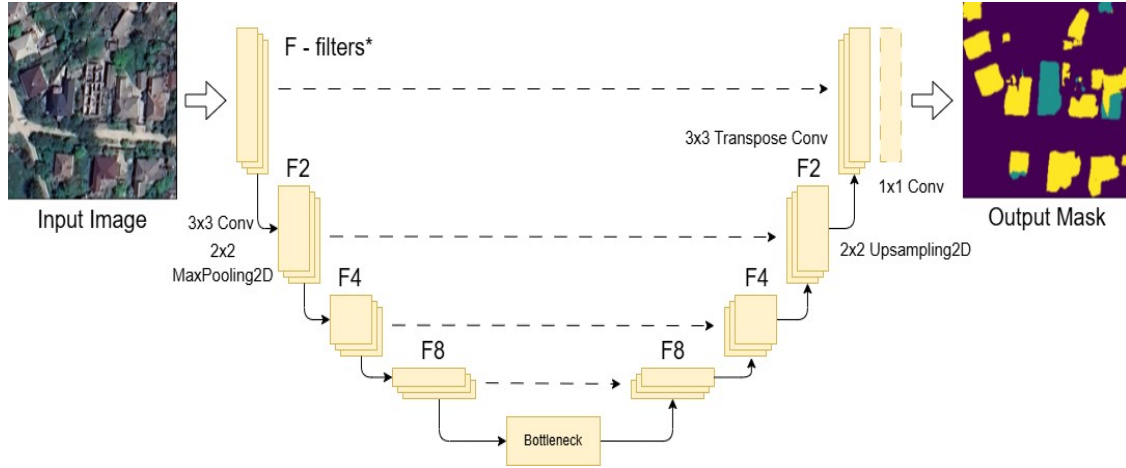
**Problem statement.** The purpose of this paper is to improve the efficiency of the U-Net convolutional network model by implementing an ASPP layer instead of a bottleneck to increase the accuracy of segmentation of buildings damaged as a result of hostilities in satellite images.

## 2. U-Net convolutional network architecture with atrous spatial pyramid pooling layer

In the framework of a previous study [12], a classical U-Net architecture model were built, originally developed by Olaf Ronneberger, Philipp Fischer, Thomas Brox [13], which is based on a fully convolutional neural network [14] with appropriate changes in the structure. The model was used for semantic segmentation of buildings damaged as a result of hostilities in satellite images, had 64 filters, and was trained on a smaller dataset (50 images with dimensions of 256 by 256 pixels) for 25 epochs. As a result, it was possible to achieve an overall accuracy of 84.21%, as well as corresponding IoU indicators of 45.83% for damaged and 49.14% for intact buildings on the evaluation dataset.

In turn, the constructed model had difficulties in determining buildings of different sizes and shapes, in some images, homestead plots (which can often be found in the private sector of Ukrainian cities and towns) could be identified as buildings. It is worth considering the peculiarities of damage as a result of hostilities, which differ from damage caused by natural phenomena such as hurricanes or floods. There was a need to improve the segmentation capabilities of the model by changing its architecture, or by using other processing methods.

Figure 1 shows the architecture of the model for segmenting damaged buildings studied in a previous work.



**Figure 1:** The general architecture of U-Net used in the study.

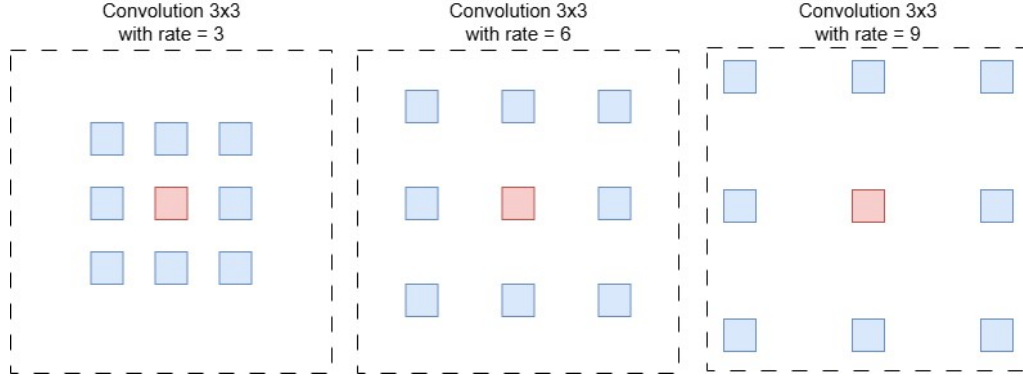
The classic U-Net architecture consists of two main parts – Contracting and Expanding paths, which are necessary for appropriate feature selection and image segmentation, which in turn is similar to the principle of operation of encoder-decoder models.

In addition, the model has various components, such as: Convolutional Block (a sequence of 3x3 convolution layers and activation functions), Encoder Block (a component of Contracting Path, which uses Convolutional Block, 2x2 pooling layer and Dropout), Decoder Block (an analogue of Encoder Block, which uses upsampling layers, concatenation of previous layers and corresponding Dropout), as well as Bottleneck, which is the narrowest point of the model with the lowest image resolution but the largest number of channels.

In this study, we propose the use of Atrous Spatial Pyramid Pooling to potentially improve the model by applying multiple pooling layers with different distance rates, which allows selecting the necessary features of objects in the image at different levels of scalability. The use of the ASPP module instead of the bottleneck will reduce the computational complexity of the network and improve image segmentation by using atrous convolutions.

Typically, the use of deep convolutional neural networks is successfully applied to most segmentation tasks due to their fully connected nature, although constant repetition of max-pooling and striding operations significantly reduces spatial resolution, and also creates a lot of parameters to calculate, which in turn significantly complicates the model and leads to additional resource consumption and time for training. The use of deconvolution layers can solve some of the problems, but requires even more memory and time.

Figure 2 shows an example of a visual representation of the atrous convolutions kernel (in other works [15] a similar principle is also given as dilated convolutions) with different parameters of the distance between the filters.



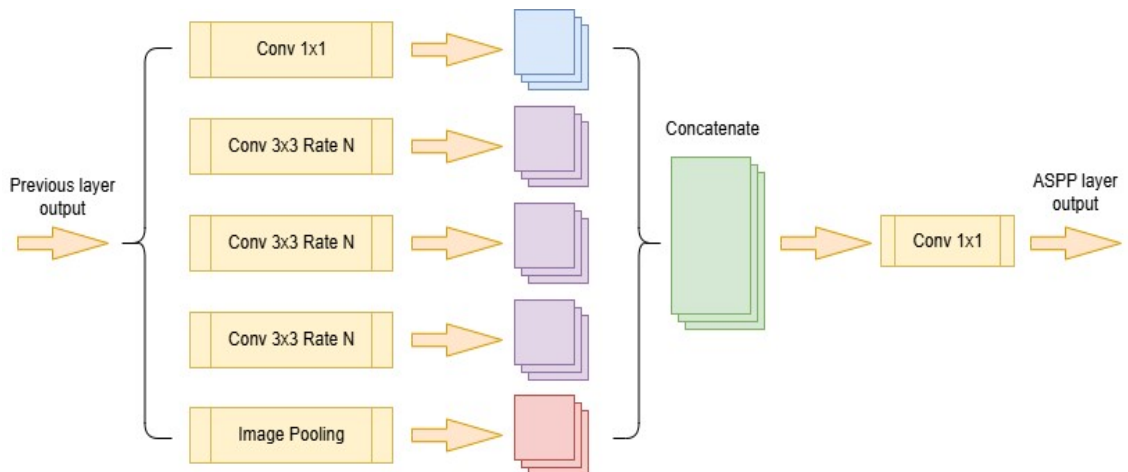
**Figure 2:** Example of atrous convolutions with different rate.

In turn, the use of atrous convolution, proposed by the authors [6, 15] allows to calculate the output data of the layers with any resolution, and can be used both after training the network and implemented together with it as well. As can be seen in the figure, if in the usual convolution operation, the filter has a fixed size, and slides over the input feature map, multiplying the values to calculate the output value, then in the case of atrous convolution the filter has “gaps”. Thus, the filter becomes larger and is able to cover more of the receptive field, which is respectively determined by the size of the filter, but at the same time this does not lead to an increase in the number of parameters for calculation, since only non-zero values are taken into account, which replace the gaps.

Spatial Pyramid Pooling was used in the original work [5] in R-CNN to eliminate the need to train a neural network on fixed-size images by using resampling of convolutional features at the same scale, which inspired the authors to create a modified version – Atrous Spatial Pyramid Pooling. ASPP uses multiple parallel convolutional layers with atrous convolutions implemented at different sampling rates. ASPP extracts features for each sampling rate, which are processed in separate layers, and then combined into one to obtain the final result [6].

As can be seen, the step-by-step application of several filters with different distances allows to significantly reduce the need for a larger number of operations (by reducing the parameters) with better results in detecting both low-level and high-level features.

The proposed ASPP module for modifying U-Net is shown in Figure 3.



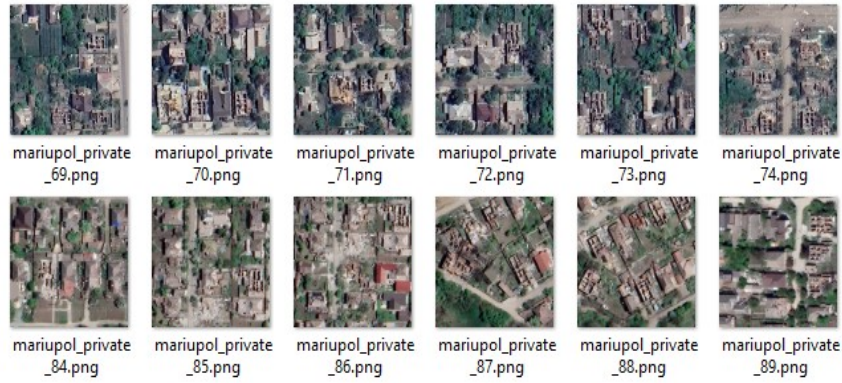
**Figure 3:** ASPP layer structure for proposed model.

The data from the Contracting Path is fed into the ASPP module, which applies a stepwise 1x1 convolution, 3x3 convolution with appropriate filter spacing (atrous convolutions), and a pooling process, followed by concatenating all previous layers into one. At the end of the module, a 1x1 convolution is performed and the processed data is fed into the Expanding Path. This allows the

model to receive more information from different scales, which in turn can improve the model's ability to detect the shape of an object in an image more accurately, which is important when segmenting damaged buildings.

### 3. Dataset, pre-processing and implementation of augmentations

The model training set was created using Google Earth. Images of the private sector of the city of Mariupol were used, which were taken around May 2022. The dataset contains 100 satellite images with dimensions of 512 by 512 pixels, and contains about more than 2500 unique instances of buildings, as well as 100 images of segmentation masks for each image (Figure 4).



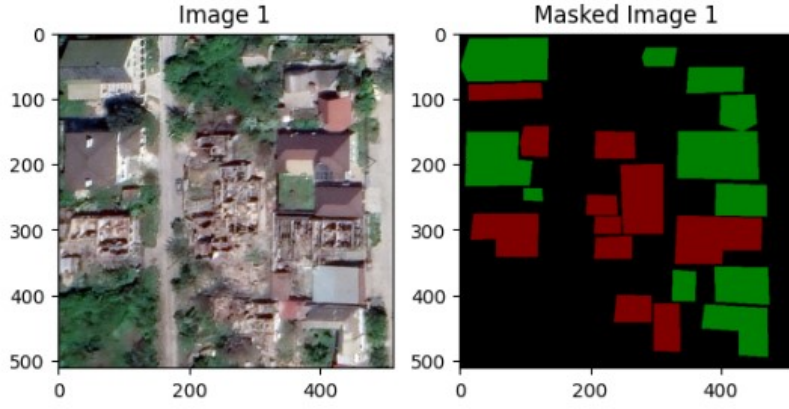
**Figure 4:** Images samples from dataset. Imagery: Google, ©2025 Airbus.

As can be seen, each image contains dozens of buildings of different shapes and scales, as well as other objects that pose significant obstacles to detection, such as roads, trees, small architectural forms, etc. This study considers the segmentation of damaged and intact buildings, without segmenting more complex and diverse objects to simplify the task.

For the segmentation task, three classes were defined, namely: background (pixels that do not belong to buildings), normal (pixels that belong to buildings that are not visually damaged), and damaged (damaged or destroyed). It is worth noting that this approach does not allow to determine the nature of the damage inside the building, therefore, in order to improve the model performance and correctly classify pixels, some buildings that were probably damaged were assigned to the normal class if the significant damage caused by the fighting could not be visually confirmed, or this damage is not significant in the context of this study.

An example of annotated images is shown in Figure 5. Black pixels indicate background, while green and red pixels indicate normal and damaged classes, respectively. The free to use Labelme software was used in the annotation process [16].



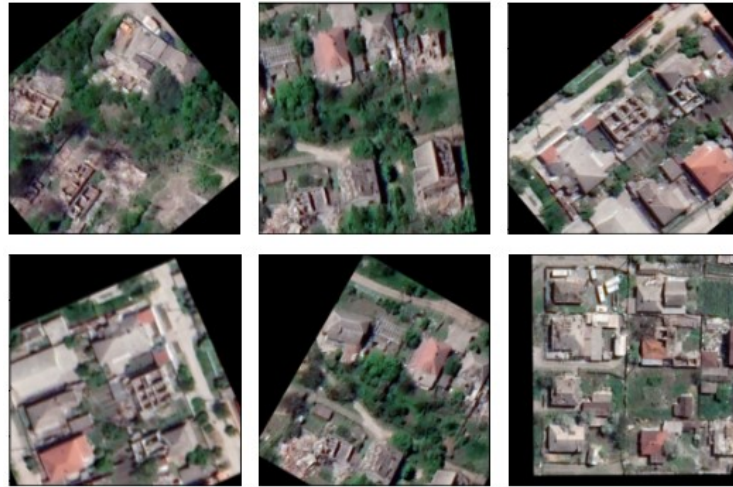


**Figure 5:** Original image and its segmented mask. Imagery: Google, ©2025 Airbus.

Image preprocessing included sharpening (reducing blur), normalization, and dimensionality rescaling as needed (if the model input is smaller than the original image size). Augmentation was also applied using the Albumentations library [17], which increased the variability and ability of the model to learn from a limited dataset.

Figure 6 shows an example of the applied augmentations. The following augmentations configurations were used in the process.

- shift\_limit=0.3, scale\_limit=0.1, rotate\_limit=270
- shift\_limit=0.2, scale\_limit=0.05, rotate\_limit=90, blur\_limit=3
- shift\_limit=0.2, scale\_limit=0.15, rotate\_limit=180



**Figure 6:** Applied augmentations to various images. Imagery: Google, ©2025 Airbus.

The final step was to create a function to generate weights for the image samples that were applied to each existing feature in the dataset (including the corresponding segmentation masks). This approach allows using weights for each pixel of the corresponding class to balance the existing classes in the dataset.

#### 4. Metrics and functions

The U-Net architecture involves the use of a Softmax function (also known as a normalized exponential function) in the last layer, which allows the original data to be transformed into a probability distribution. In this case, this means that each pixel will be assigned a corresponding probability value for belonging to each class, and the total sum will be equal to one, after which the

functions of obtaining the largest value can be applied to finally determine the most likely class to which a particular pixel belongs.

The cost function used is the categorical cross entropy (CCE), which is often used in semantic segmentation problems. CCE calculates the “distance” between the actual class distribution and the corresponding prediction; a lower score indicates a greater degree of agreement between the predicted value and reality. Since CCE calculates the score based on probability distributions, Softmax is used before applying the loss function.

Formulas 1 and 2 reflect the corresponding Softmax and CCE functions [18-20]:

$$SoftMax(image_c) = \frac{\exp(image_c)}{\sum_{j=1}^k \exp(image_c)} \quad (1)$$

where  $c$  is the current class in the range from 1 to  $k$ ,  $j$  is the iteration from 1 to  $k$ ,  $image$  is the output data (image), and  $\exp$  is the exponent of the corresponding set of values.

$$CCE = - \sum_{c=1}^k ground\_truth_c \times \log(image_c) \quad (2)$$

where  $c$  is the current class in the range from 1 to  $k$ ,  $image$  is the original data (image), and  $ground\_truth$  is the segmentation mask for the corresponding image.

Since the work considers semantic image segmentation, Intersection over Union (IoU) was chosen as a metric for calculating the effectiveness of the model, which takes into account the ratio of the number of pixels of the original image with the corresponding segmentation map. During training and testing, IoU is calculated for two main classes – intact and damaged buildings, as well as the average IoU value between all classes, which allows for a more complete understanding of some models [21].

Formulas 3 and 4 for IoU are given below [22-24].

$$IoU_c = \frac{TP_c}{TP_c + FP_c + FN_c} \quad (3)$$

where  $c$  is the current class for calculation,  $TP$  is the ‘true positive’ value,  $FP$  is the ‘false positive’ value,  $FN$  is the ‘false negative’ value of the pixels.

$$IoU_{mean} = \frac{1}{C_{max}} \sum_c IoU_c \quad (4)$$

where  $c$  is the current class to calculate,  $C_{max}$  is total classes,  $IoU_c$  is the calculated IoU value for class  $c$ .

In turn, the use of weights allows the model to focus on more important classes, for example, on building classes instead of background. The corresponding function values are multiplied by weights, which at the output gives an adjusted result, for example, the loss function takes into account the different importance between classes, increasing the loss for targeted classes. The selection of weights is quite a complex task and may require additional research.

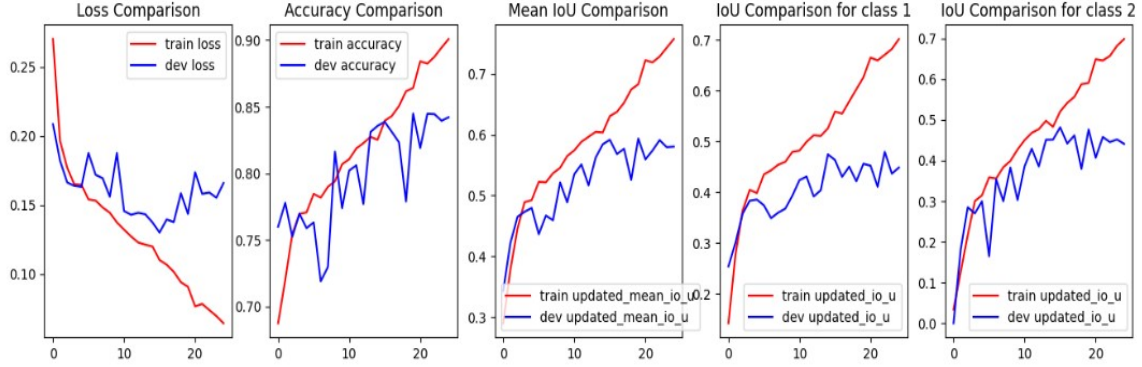
## 5. Models training and testing. Comparative analysis

The model building and training was performed in Python using the Tensorflow and Keras libraries [25], as well as the Google Colab cloud machine learning service [26] using an NVIDIA T4 GPU.

At the beginning of the study, two baseline models were compared: U-Net with a conventional architecture and with ASPP as a bottleneck. Training was performed on 256 by 256 pixels images due to the memory limitations of the GPU. Appropriate augmentations were applied to the dataset, which increased the image volume by a factor of five, of which 70% was divided for model training,

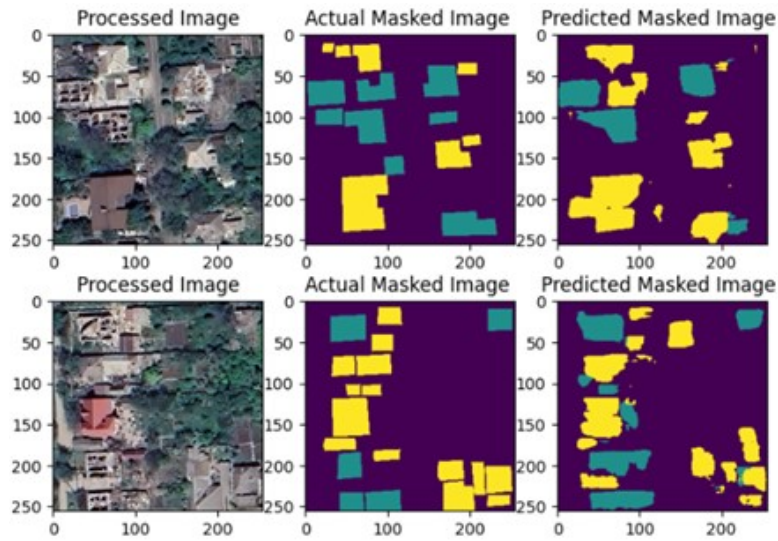
15% for model evaluation, and 15% as test images that were not used in training or evaluation, which is necessary to obtain more objective results regarding the quality of the model.

Figure 7 shows charts with the results of training and evaluation of the baseline model using ASPP as a bottleneck. Overall, the base model shows result comparable to the model without ASPP, but has lower losses and higher average IoU.



**Figure 7:** Training results of the base model with ASPP bottleneck.

Figure 8 shows some segmented images from the testing set. These images were not included in the model training, so the results represent a realistic assessment of the model’s performance on unique data.



**Figure 8:** Some of segmented images on test data from base model. Imagery: Google, ©2025 Airbus.

It is possible to note a potentially positive dynamics of pixel classification into the corresponding classes, although the model still has insufficient data on some objects, for example, adjacent areas. Since the task is to segment two main classes: damaged and intact buildings, the model has difficulty recognizing objects that have “transitional” features between these two classes, as well as if the environment has features of buildings – as in the case of construction debris or completely destroyed buildings. An acceptable solution to solve such features may be to expand the problem, with the segmentation of intermediate classes that will characterize the degree of damage.

As a result, 2 basic U-Net models (classical and with ASPP) were compared, as well as 3 best models with other parameters. Table 1 shows the basic configuration of each model that was tested in the study.



**Table 1**  
Configurations of built models

Model	Weights	Augmentations	Epochs	ASPP config
Base model without ASPP	1.0, 3.0, 2.0	x5	25	-
Base model with ASPP	0.5, 1.5, 1.5	x3	25	1, 6, 12, 18
Model 1	1.0, 3.0, 2.0	x5	25	1, 6, 12, 18
Model 2	1.0, 4.0, 4.0	x5	30	1, 6, 12, 18
Model 3	1.0, 4.0, 4.0	X5	30	1, 3, 6, 9

The results of the comparative analysis are given in Table 2.

**Table 2**  
Comparative analysis of built models

Model	Loss		Accuracy		IoU Damaged		IoU Normal		Mean IoU	
	Val	Test	Val	Test	Val	Test	Val	Test	Val	Test
Base model without ASPP	0.4023	-	0.8421	-	<b>0.4583</b>	-	<b>0.4914</b>	-	0.5261	-
Base model with ASPP	<b>0.1659</b>	<b>0.284</b>	<b>0.8521</b>	<b>0.838</b>	0.4484	0.427	0.44	0.411	<b>0.58</b>	<b>0.543</b>
Model 1*	0.207	0.411	0.8186	0.807	0.4216	0.418	0.451	0.448	0.566	0.538
Model 2	<u>0.174</u>	<u>0.381</u>	0.8	0.783	0.449	0.422	0.415	0.423	0.551	0.522
Model 3	0.215	0.455	<u>0.831</u>	<u>0.814</u>	<u>0.451</u>	<b>0.431</b>	<u>0.457</u>	<b>0.451</b>	<u>0.572</u>	<u>0.541</u>

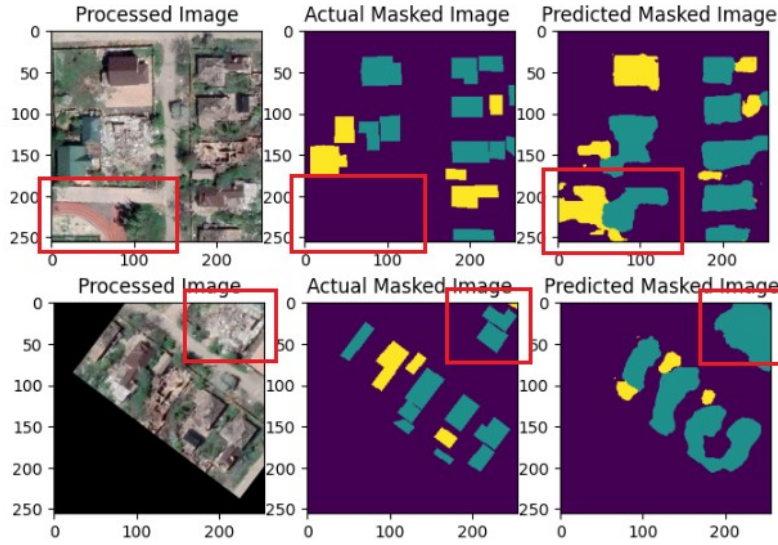
Thus, it is possible to note the improvement of the main metrics such as loss, accuracy and mean IoU in the baseline model using ASPP. At the same time, the IoU value for damaged and intact buildings is lower. This can be explained by the fact that the model reduced the probable area of classification of buildings “above” the background class, which partially reduced the IoU indicator but increased the overall average IoU value. As a result, the baseline model has 0.2364 less losses on the evaluation set, increased accuracy by 0.1 and the average IoU value by 0.0539 compared to the classical model, which indicates the positive impact of ASPP on the segmentation of damaged buildings.

Model 3 has the following better performance, such as increased IoU values for damaged and intact buildings, as well as an average IoU value of 0.541, close to that of the baseline model with ASPP, which may indicate a positive impact of reducing the atrous rate in the context of the task.

It is worth noting that Model 1 showed the best results among all models at 23 epochs on the evaluation dataset: 0.4583 IoU for damaged, 0.5147 IoU for intact, with a 0.6072 average IoU value among the three classes, making this model potentially the best among those built.

Despite the improvements, the model still requires addressing other inaccuracies, such as the possibility of confusing objects not related to buildings (running tracks, sports fields, the pixels of

which may coincide with pixels that are characteristic of buildings), or the inability to distinguish completely destroyed objects from rubble (Figure 9).



**Figure 9:** Example images on test data from base model. Imagery: Google, ©2025 Airbus.

This can be addressed in several ways, including using a larger dataset, expanding the classes for segmentation (e.g., adding the degree of damage to buildings), or changing the approach to data annotation, as the impact of annotation on the model's accuracy in detecting damage to buildings and the environment due to combat operations requires more detailed investigation, which will be performed in future studies.

## 6. Conclusions

The paper considered improving the efficiency of the U-Net CNN model using atrous spatial pyramid pooling to increase the accuracy of segmentation of damaged buildings. The implemented ASPP module was used as a bottleneck, which allowed not only to cover more image features at different levels, but also to reduce the computing power required for training.

For the study, the dataset was expanded to 100 original images, and various augmentations were applied to increase the variability of the dataset, which generally positively affected the model's ability to train on a limited number of images. Weights were also generated for each image, which gave more weight to pixels in specific classes to balance out the data that predominated.

As part of the study, a baseline model was built using ASPP, and the 3 best ones were selected using different parameters. It was determined that the implementation of ASPP has a positive effect on the training efficiency and the quality of the final results. According to the comparative analysis, it was possible to increase the average IoU value on the validation dataset by 5.39% for the baseline model using ASPP. At the same time, Model 3 has higher IoU values for both classes of buildings, which may indicate a positive effect of reducing the distance between the filters. It is also worth noting the evaluating results of Model 1, which received the best results among all models at 23 epochs: 45.83% and 51.47% IoU for damaged and intact buildings, respectively, as well as an average IoU of 60.72%, which makes this model potentially the best among the others.

The positive impact of ASPP application allows using this module in further studies aimed at reducing the impact of other features of segmentation of damaged buildings as a result of hostilities, such as the difficulties of detecting destroyed buildings from rubble and of accurately determining the shape of damaged buildings and others.

## Declaration on Generative AI

The authors have not employed any Generative AI tools.

## References

- [1] J. Gu, Z. Xie, J. Zhang, X. He, Advances in rapid damage identification methods for post-disaster regional buildings based on remote sensing images: A survey, *Buildings* 14 (2024) 898. doi:10.3390/buildings14040898.
- [2] C. L. Moreno González, G. A. Montoya, C. Lozano Garzón, Toward reliable post-disaster assessment: Advancing building damage detection using You Only Look Once convolutional neural network and satellite imagery, *Mathematics* 13 (2025) 1041. doi:10.3390/math13071041.
- [3] J. Liu, Y. Luo, S. Chen, J. Wu, Y. Wang, BDHE-Net: A novel building damage heterogeneity enhancement network for accurate and efficient post-earthquake assessment using aerial and remote sensing data, *Appl. Sci.* 14 (2024) 3964. doi:10.3390/app14103964.
- [4] T.-Y. Lin, P. Dollár, R. Girshick, K. He, B. Hariharan, S. Belongie, Improved feature pyramid networks for object detection, in: *Proc. IEEE Conf. Comput. Vis. Pattern Recognit., CVPR 2017, Honolulu, HI, 2017*, pp. 936–944. doi:10.1109/CVPR.2017.106.
- [5] K. He, X. Zhang, S. Ren, J. Sun, Spatial pyramid pooling in deep convolutional networks for visual recognition, *IEEE Trans. Pattern Anal. Mach. Intell.* 37 (2015) 1904–1916.
- [6] L.-C. Chen, G. Papandreou, I. Kokkinos, K. Murphy, A. L. Yuille, DeepLab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected CRFs, *IEEE Trans. Pattern Anal. Mach. Intell.* 40 (2018) 834–848.
- [7] L.-C. Chen, Y. Zhu, G. Papandreou, F. Schroff, H. Adam, Encoder-decoder with atrous separable convolution for semantic image segmentation, *CoRR* (2018). doi:10.48550/arXiv.1802.02611.
- [8] M. Yu, W. Zhang, X. Chen, Y. Liu, J. Niu, An end-to-end atrous spatial pyramid pooling and skip-connections generative adversarial segmentation network for building extraction from high-resolution aerial images, *Appl. Sci.* 12 (2022) 5151. doi:10.3390/app12105151.
- [9] L. Hu, X. Zhou, J. Ruan, S. Li, ASPP+LANet: A multi-scale context extraction network for semantic segmentation of high-resolution remote sensing images, *Remote Sens.* 16 (2024) 1036. doi:10.3390/rs16061036.
- [10] Y. Miao, S. Jiang, Y. Xu, D. Wang, Feature residual analysis network for building extraction from remote sensing images, *Appl. Sci.* 12 (2022) 5095. doi:10.3390/app12105095.
- [11] Z. Wang, Y. Chen, F. Wang, Q. Bao, Improved Unet model for brain tumor image segmentation based on ASPP-coordinate attention mechanism, in: *Proc. 5th Int. Conf. Big Data Artif. Intell. Softw. Eng., ICBASE 2024, Wenzhou, China, 2024*, pp. 393–397. doi:10.48550/arXiv.2409.08588.
- [12] A. Gozhyj, I. Kalinina, V. Dymo, Application of convolutional neural networks for detection of damaged buildings, *CEUR-WS* 3711 (2024) 15–27. URL: <http://CEUR-WS.org/Vol-3711/paper2.pdf>.
- [13] O. Ronneberger, P. Fischer, T. Brox, U-Net: Convolutional networks for biomedical image segmentation, 2015. URL: <https://arxiv.org/pdf/1505.04597.pdf>.
- [14] J. Long, E. Shelhamer, T. Darrell, Fully convolutional networks for semantic segmentation, in: *Proc. IEEE Conf. Comput. Vis. Pattern Recognit., CVPR 2015*. doi:10.48550/arXiv.1411.4038.
- [15] F. Yu, V. Koltun, Multi-scale context aggregation by dilated convolutions, in: *Proc. Int. Conf. Learn. Representations, ICLR 2016*. doi:10.48550/arXiv.1511.07122.
- [16] Labelme, Image polygonal annotation with Python, 2024. URL: <https://github.com/labelmeai/labelme>.
- [17] A. Buslaev, A. Parinov, E. Kvedchenya, V. Iglovikov, Albumentations: fast and flexible image augmentations, in: *Proc. IEEE Conf. Comput. Vis. Pattern Recognit., CVPR 2018*. URL: <https://arxiv.org/abs/1809.06839>.
- [18] C. M. Bishop, *Pattern Recognition and Machine Learning*, Springer, New York, NY, 2006. ISBN: 0-387-31073-8.

- [19] A. Gozhyj, V. Nechakhin, I. Kalinina, Solar Power Control System based on Machine Learning Methods. 23-26 September (2020) Corpus ID: 231714812. Conference: 2020 IEEE 15th International Conference on Computer Sciences and Information Technologies (CSIT). doi:10.1109/CSIT49958.2020.9321953.
- [20] S. Babichev, B. Durnyak, V. Zhydetskyy, I. Pikh, V. Senkivskyy, Techniques of DNA Microarray Data Pre-processing Based on the Complex Use of Bioconductor Tools and Shannon Entropy. CEUR Workshop Proceedings 2353, Zaporizhzhia (2019) pp. 365-377, <https://ceur-ws.org/Vol-2353/paper29.pdf>.
- [21] P. Bidiuk, A. Gozhyj, Z. Szymanski, I. Kalinina, V. Beglytsia, The Methods Bayesian Analysis of the Threshold Stochastic Volatility Model. Journal 2018 IEEE 2nd International Conference on Data Stream Mining and Processing, DSMP '2018, Lviv. (2018) 70-74. doi: 10.1109/DSMP.2018.8478474
- [22] T. A. Taha, A. Hanbury, Metrics for evaluating 3D medical image segmentation: analysis, selection, and tool, BMC Med. Imaging 15 (2006) 29. doi:10.1186/s12880-015-0068-x.
- [23] V. Andrunyk, A. Vasevych, L. Chyrun, N. Chernovol, N. Antonyuk, A. Gozhyj, V. Gozhyj, I. Kalinina, M. Korobchynskyi. Development of Information System for Aggregation and Ranking of News Taking into Account the User Needs (2020), <https://ceur-ws.org/Vol-2604/paper74.pdf>.
- [24] V. Senkivskyy, I. Pikh, N. Senkivska, I. Hileta, O. Lytovchenko, Y. Petyak, Forecasting Assessment of Printing Process Quality. Journal of Graphic Engineering and Design (2020) 11(1), pp. 27-35, [https://doi.org/10.1007/978-3-030-54215-3\\_30](https://doi.org/10.1007/978-3-030-54215-3_30).
- [25] M. Abadi, A. Agarwal, et al., TensorFlow: Large-scale machine learning on heterogeneous distributed systems, 2016. arXiv:1603.04467.
- [26] Colaboratory, Google, 2024. URL: <https://research.google.com/colaboratory/faq.html>.