

Neural network-based automation of spatial solutions in modern interior design

Pavlo Kruk^{1,†}, Oleksii Matsiievskiy^{1,†}, Igor Achkasov^{1,†}

¹ Kyiv National University of Construction and Architecture, 31, Air Force Avenue, Kyiv, 03037, Ukraine

Abstract

The work is devoted to the automation of spatial solutions in modern interior design using deep learning. An approach is proposed in which a neural network performs semantic segmentation of the room plan and predicts the placement of the main interior objects, taking into account planning constraints and stylistic preferences. Architecturally, the core of the system is a U-Net segmentation model with an extended input: in addition to the room mask, binary channels for doors and windows, a continuous distance map to the walls, and one-hot encoding of style (Minimalistic, Cozy, Modern) are provided. To enhance generalization ability, a combined loss function (weighted cross-entropy together with Soft Dice for the foreground) and augmentations relevant to the plans are used. The alternative DeepLabv3+ (ResNet-50) architecture is considered separately as a strong comparative basis for thin boundary structures.

Keywords

deep learning; interior design; semantic segmentation; furniture layout; generative design; U-Net; DeepLabv3+; automation of spatial solutions

1. Introduction

Modern interior design [1] is increasingly taking place under conditions of tight deadlines, highly variable customer requirements, and the need to quickly review alternative planning scenarios. Even for relatively simple spaces, designers must coordinate room geometry, window and door placement, functional zoning, ergonomic standards, communications, and stylistic preferences [2]. Traditional work processes – a combination of manual sketching, CAD tools [3], and a set of empirical rules – support individual craftsmanship well, but create bottlenecks when it is necessary to quickly review dozens of layouts and justify the choice [4]. In this context, methods that can automate routine steps, ensure consistency in decisions, and at the same time leave room for authorial control become particularly relevant [5].

Deep learning offers a natural path to such automation, as it allows the task of furniture placement to be interpreted as a class prediction task for each pixel of the plan [6]. Unlike hard-coded heuristics, neural networks learn spatial patterns and relationships between elements of the scene [7-8].

ITTAP'2025: 5th International Workshop on Information Technologies: Theoretical and Applied Problems, October 22-24, 2025, Ternopil, Ukraine, Opole, Poland

^{1*} Corresponding author.

[†] These authors contributed equally.

✉ kruk_pm-2023@knuba.edu.ua (P. Kruk); matsiievskiyolexiy@gmail.com (O. Matsiievskiy); achckasov.i@ukr.net (I Achkasov)

ORCID 0000-0002-6786-452X (P. Kruk); 0009-0008-2341-8166 (O. Matsiievskiy); 0000-0002-7049-0530 (I Achkasov)



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

At the same time, practical effectiveness depends not only on the choice of architecture, but also on the correct setting of inputs: the model should receive exactly those signals that the designer uses in their work. In this work, we formalize this approach and propose a compact, reproducible system called “Neural Network-Based Automation of Spatial Solutions in Modern Interior Design [9-10]”. The key idea is to use explanatory features at the model input. In addition to the room mask, binary information about doors and windows, a continuous map of distances to walls, and stylistic context in one-hot format (Minimalistic, Cozy, Modern) are provided [11]. Thus, the neural network “sees” not just an image, but a structured set of clues that in practice form the layout: where the openings are located, where the periphery and center are, and what compositional scenario is expected. U-Net [12] with three levels of encoder/decoder and skip connections was chosen as the base architecture; for finer boundary structures, a comparative baseline DeepLabv3+ with ResNet-50 [13] and ASPP [14] is considered. Training is conducted with a combined loss function: weighted cross-entropy (background is underestimated, small classes are amplified) combined with Soft-Dice for the foreground, which increases sensitivity to thin and small objects [15-16].

To reduce the burden of manual labeling and speed up iterations, we generate synthetic examples with built-in semantic rules (table by the window, sofa and TV on opposite walls, rug in the center, etc.) and anticipate further fine-tuning on real user plans. The system is accompanied by a simple interactive interface that allows you to upload a plan, select a style, and obtain a color map with a legend and basic metrics; all artifacts (convergence graphs, confusion matrix, class-wise IoU) are automatically generated for inclusion in the publication.

Our contribution lies in combining explanatory multi-channel input, compact segmentation architecture, and a transparent training/evaluation pipeline focused on practical applicability for early design stages. It has been demonstrated that even with moderate computational resources, the system consistently “pulls” the scene skeleton (perimeter, doors, windows, central dominants) and creates a basis for further enhancement of complex furniture classes through data balancing and retraining on real drawings.

2. Related Work

The automation of spatial solutions in interior design has historically relied on rule-based and parametric approaches in CAD/CAAD environments. In such systems, the layout is determined by sets of deterministic constraints — passability, door opening areas, minimum clearances, permissible furniture dimensions — and a search is performed in the space of permissible configurations. The advantage is complete controllability and transparent validation, but the complicated representation of “soft” criteria, stylistic preferences, and local exceptions leads to uniformity of results and high costs for manual sorting of options.

With the advent of deep learning, methods of semantic processing of plans have attracted considerable attention, where the layout task is interpreted as pixel-wise class prediction. Thanks to symmetrical encoder-decoder paths and skip connections, U-Net family architectures have become the standard for marking up two-dimensional plans and drawings. Further development is associated with multi-scale context aggregation, extended convolutions, and the ASPP module, which is enshrined in approaches such as DeepLabv3+. Such models are better at capturing thin lines and boundaries, which is especially relevant for doors and windows, but require high-quality data and well-thought-out input features to avoid semantic mergers of objects with similar geometry.

At the same time, scene-oriented representations are being developed: the interior is described as a graph of objects with distance, orientation, and compatibility relationships. These approaches are convenient for further parametric editing and generative synthesis, but

their practical effectiveness on “raw” plans often depends on reliable initial segmentation or detection, which brings us back to the task of pixel interpretation.

A key problem for all learning approaches is the lack of annotated data. Therefore, synthetic datasets, domain adaptation, and targeted augmentations are actively used. Synthetics allow us to embed “common sense” spatial relationships — the desk gravitating toward the window, the sofa opposite the TV, the rug in the center — and quickly increase the variety of scenarios. However, transfer to real plans usually requires additional training and simple tools for manual validation of results.

An important component of modern solutions is interactivity and transparency, where the system not only outputs a mask, but also visualizes confidence, errors, and class legends. This simplifies integration into the studio's workflow, where the designer, while maintaining creative control, quickly weeds out unsuccessful configurations and fixes successful ones.

3. Problem Formulation

The task of semantic interior layout is considered as a pixel-based multi-class classification. For a room with dimensions $H \times W$ the input tensor is formed $X \in R^{H \times W \times C}$ with an extended set of features: binary mask of the room, binary channels of doors and windows, continuous map of distance to walls $d_{wall} \in [0, 1]$, as well as one-hot encoding style $s \in \{0, 1\}^s$ (Minimalistic, Cozy, Modern), repeated in spatial coordinates. Collectively $C = 4 + s$ channels. Model f_{θ} (U-Net also DeepLabv3+) reflects X in $\log Z = f_{\theta}(X) \in R^{H \times W \times K}$, K — number of classes (background class and at least 12 types of furniture). Probabilities are calculated as $P = \text{softmax}(Z)$, final marking — $\hat{Y} = \text{argmax}_k P[:, :, k]$.

Training is conducted under supervision with a reference mask $Y \in \{0, \dots, K-1\}^{H \times W}$. To account for the imbalance in class areas and enhance sensitivity to thin objects, the combined loss function is optimized.

$$L(\theta) = a C E_w(Y, Z) + (1 - a) (1 - \text{Dice}_{fg}(P, \text{onehot}(Y))) \quad (1)$$

Where $C E_w$ — weighted cross-entropy (weights are inverse logarithmic to class frequencies, background is additionally reduced), Dice_{fg} — Soft-Dice without background class, $a \in [0, 1]$ (in practical settings $a = 0.5$). Parameters θ AdamW is optimized with moderate regularization and standard augmentations that are correct for plans (90° rotations, flips, slight scaling, noise in d_{wall}).

The evaluation is performed under deferred validation. The main integral metric is the mean intersection-over-union ratio (mIoU), defined as:

$$E mIoU = \frac{1}{K} \sum_k \frac{TP_k}{TP_k + FP_k + FN_k} \quad (2)$$

where mIoU — mean intersection-over-union, computed as the average overlap between predicted and ground-truth regions across K classes; TP_k , FP_k , FN_k — numbers of true positives, false positives, and false negatives for class k , respectively. This metric evaluates segmentation quality by measuring the consistency between predicted and actual object boundaries, independent of class size. The reporting is performed in two modes: “all classes” and “no background”. Additionally, pixel accuracy is provided as:

$$PixelAcc = \frac{\sum_k TP_k}{\sum_k (TP_k + FN_k)} \quad (3)$$

where *PixelAcc* — pixel accuracy, representing the proportion of correctly classified pixels among all labeled pixels; TP_k — true positive pixels for class k ; FN_k — false negative pixels for class k . Pixel accuracy reflects overall prediction correctness without accounting for class imbalance.

In addition, standard classification metrics are reported, including Precision, Recall, and F1-score, as well as a confusion matrix normalized by rows for all foreground classes. Confidence maps are used to interpret the results through the maximum class probabilities $\max_k P[:, :, k]$ and corresponding error masks $1[\hat{Y} \neq Y]$. This approach ensures reproducibility, style parameterization, and integration into automated sketching tools.

4. Neural network model

At the heart of the proposed system is a U-Net-type segmentation network with an extended input, shown in Figure 1, which encodes spatial features relevant to the layout.

A seven-channel image is fed into the network: a room mask, binary maps of doors and windows, a continuous map of distances to walls, and three one-hot style channels (Minimalistic, Cozy, Modern). Thanks to this, the model receives not a “picture” but a structured set of hints for planning.

The architecture is U-Net in a symmetric configuration with three encoder/decoder levels ($64 \rightarrow 128 \rightarrow 256$) and a “bottleneck” on 512 channels. The encoder extracts increasingly abstract features (with max-pooling), while the decoder restores detail with transposed convolutions and skip connections that return local contours. The final head converts the features into class channels (background + 12 types of furniture).

Each block consists of two consecutive convolutions with BatchNorm and ReLU, which ensures stability and sufficient expressiveness without unnecessary complexity. Skip connections act as a “conveyor” of spatially accurate features: door/window contours, carpet edges, straight table and cabinet fronts are correctly reproduced in the upper layers.

The key to quality is correct features at the input: the distance map to the walls gives a “center-periphery” hint, doors and windows fix landmarks, and the stylistic code allows you to change compositional preferences without changing the architecture.

Training is performed using a combination of weighted Cross-Entropy (less weight for the background, more for small classes) and Soft-Dice for the foreground, which enhances sensitivity to subtle boundaries. The optimizer is AdamW. Augmentations take into account the nature of the plans: 90° rotations, horizontal/vertical flips, slight scaling, and minor noise only in the dist2wall channel — this preserves the geometric meaning of the scene.

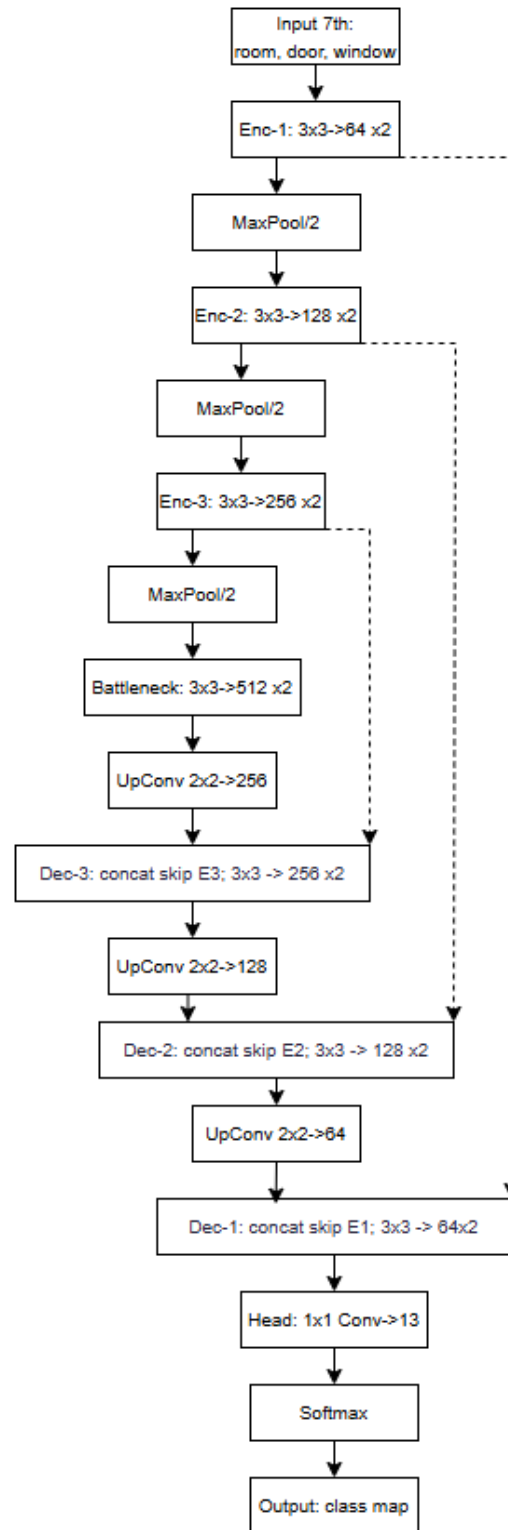


Figure 1: U-Net scheme with seven-channel input for semantic segmentation of interior plans.

For comparison of fine structures, we consider DeepLabv3+ (ResNet-50, ASPP) with the first layer adapted to 7 channels. In production scenarios, it is useful for critical boundaries, but in most prototype cases, the compact and fast U-Net remains the baseline.

5. Software implementation

The software implementation is built on Python 3.10+ / PyTorch, focused on reproducibility and direct inclusion of artifacts in the article. The model input is formed as a seven-channel image: a room mask, binary maps of doors and windows, a continuous map of distances to walls, and three one-hot style channels (Minimalistic, Cozy, Modern). The base architecture is U-Net with three encoder/decoder layers and skip connections; the classification head returns 13 logits (background + 12 furniture types). The loss function combines weighted cross-entropy (background underestimated; rare classes reinforced) and Soft-Dice only for the foreground, which increases sensitivity to thin and small objects. The optimizer is AdamW with moderate regularization; augmentations are adapted to the plans (90° rotations, flips, slight scaling, weak noise only in the dist2wall channel).

Figure 2 shows a monotonic decrease in both training and validation losses (from approximately ~ 1.75 to ~ 1.30), with no noticeable difference between the curves. This indicates stable convergence and no early overfitting under low resolution (160×160) and multi-class labeling conditions. The generalized metrics confirms this trend: mIoU(all) increases to ~ 0.20 , mIoU without background gradually gains, and pixel accuracy reaches ~ 0.7 at the end of the trial cycle. For a prototype with a limited number of epochs, this is a typical scenario: the model first “learns” the scene framework, then gradually improves the quality by class.

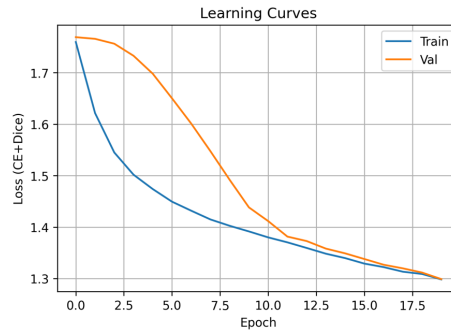


Figure 2: Learning Curves CE+Dice

The normalized confusion matrix for the foreground, shown in Figure 3, shows strong diagonals for doors and windows (values close to 1.00), which is logical: these objects have distinct cues at the entrance and clear geometry. The carpet also stands out with a relatively high value (≈ 0.75). At the same time, there are characteristic cross-errors between rectangular furniture of similar size and location: the bed is partially perceived as a wardrobe; the bookshelf as a desk; the TV has spillover into adjacent rectangular classes. This pattern is consistent with expectations for imbalanced data with background dominance and classes with similar shapes.

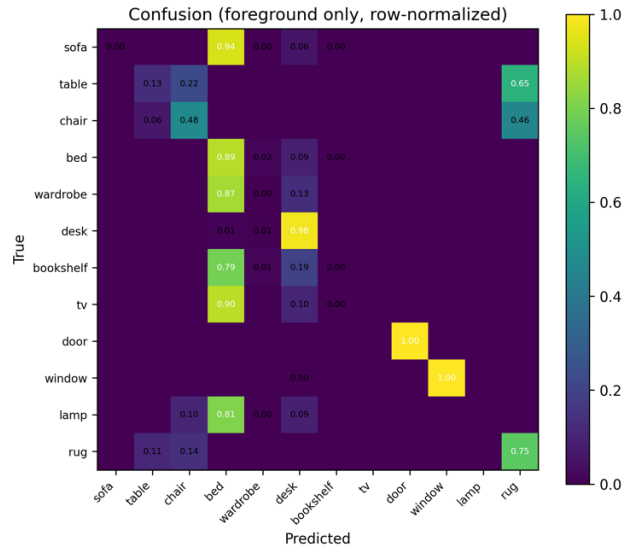


Figure 3: Matrix of confusion

Figure 4 shows the IoU for each class without background in descending order with an additional “support” curve (number of pixels). The best values are demonstrated by doors (~0.59) and windows (~0.54), followed by carpets (~0.31) and desks (~0.31). Tables, chairs, and beds have low IoUs (about 0.10, 0.08, and 0.03, respectively), and several classes with very small areas have practically no scores — a typical consequence of imbalance and a small number of epochs. The accompanying support curve indicates that some classes are objectively present in the data much less frequently or occupy a very small area, which increases the volatility of estimates.

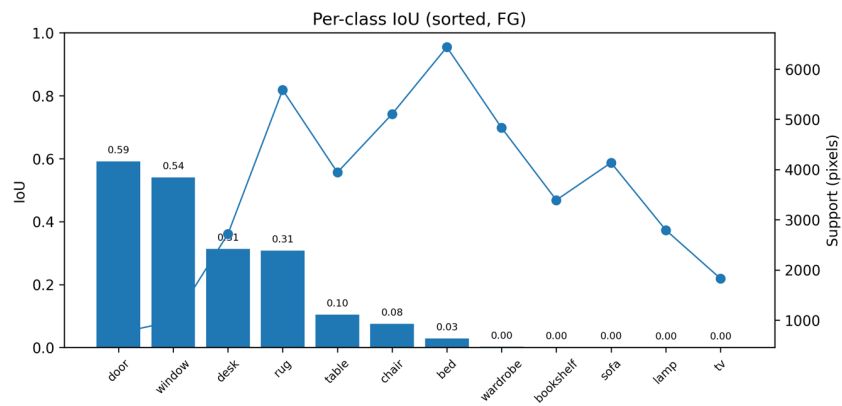


Figure 4: Per-class IoU with Support

The grid in Figure 5 illustrates the model's behavior on a typical set of scenes: the “Input” column with the selected style, the ground truth (GT) in the colors of the legend, the prediction (Pred), the confidence map (Confidence), and the error map (Error). It is clearly visible that the network “pulls” the perimeter of the room thanks to dist2wall, consistently

localizes doors and windows, as well as the central carpet. Errors are concentrated on the edges of large rectangular furniture and in small classes — on the error maps, this is manifested by bright “islands” in the corners and along the contours.

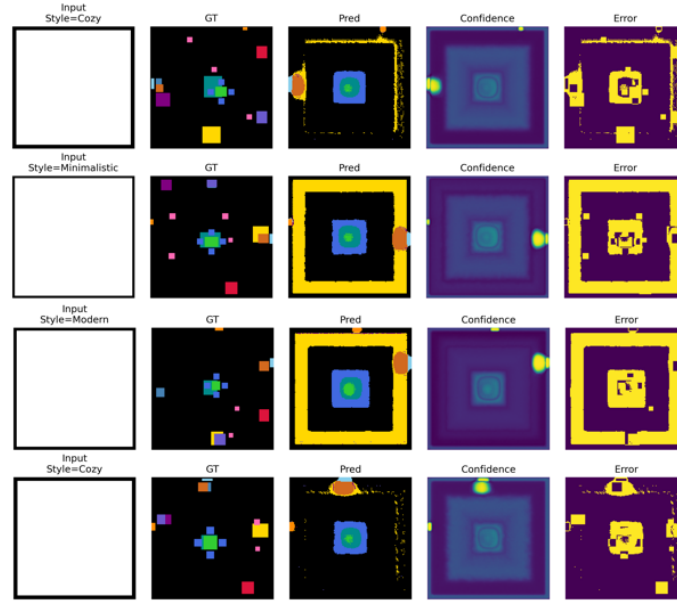


Figure 5: Qualitative Samples: Input / GT / Prediction / Confidence / Error

The selected feature set and architecture correctly “extract” the scene’s framework (perimeter, doors, windows, central landmarks) and gradually improve the quality of furniture classes. At the same time, cross-errors between similar rectangular silhouettes and low IoU of small classes indicate the need for additional measures: balancing the sample, strengthening the boundary-oriented component of losses (Lovász-Softmax/BCE-Boundary), soft post-processing (CRF/morphology) and, most importantly, retraining on real plans with a sufficient number of examples for weak classes. In a production scenario, it is also appropriate to test the alternative DeepLabv3+ head (ResNet-50, ASPP) — it usually works better with thin door/window boundaries while maintaining compatibility with the seven-channel input.

Conclusion

This paper proposes and experimentally substantiates an approach to automated support for spatial solutions in modern interior design based on deep learning. The central elements are a seven-channel explanatory input (room, door, window, dist2wall, style×3) and a compact U-Net segmentation architecture with a combined loss function (weighted Cross-Entropy + Soft-Dice for the foreground). The results obtained showed stable convergence and convincing quality for the “anchor” classes (doors, windows, carpet), confirming the feasibility of presenting precisely those features that guide practical planning. At the same time, expected limitations were identified for classes with similar rectangular geometry and for small objects, which necessitates data balancing and additional boundary enhancement mechanisms.

The limitations of the study are related to moderate image resolution, a limited number of epochs, class imbalance, and the proportion of synthetic examples. In line with this, steps are

planned to further improve quality: expanding the real sample, purposefully balancing “weak” classes, testing boundary-aware loss functions (Tversky/focal-Tversky) and options for multi-scale context aggregation (in particular DeepLabv3+ as a comparative architecture). A separate area of research is domain adaptation between synthetic plans and real drawings.

In summary, the work demonstrates a viable approach to using deep learning as a tool to support spatial decision-making in the early stages of design. The combination of explanatory input, segmentation architecture, and transparent visualizations already provides consistent, interpretable results. Further development in terms of data, losses, architectures, and, especially, visualizations and UI/UX with customizable furniture selection and room parameters has every reason to significantly increase the practical usefulness of the system for a wide range of users.

Declaration on Generative AI

During the preparation of this work, the author used OpenAI GPT-5 Thinking as a generative AI assistant for limited, non-substantive support. In line with the CEUR-WS activity taxonomy, the usage falls under editorial assistance (grammar, punctuation, and style checks for English text), programming assistance (help with code editing such as fixing minor errors, small optimizations, clarifying inline comments, and light refactoring of the provided snippets), and data support (suggestions for assembling and selecting synthetic test examples and simple augmentation schemes for training/validation).

After using the above tool, the author reviewed and edited all text, code, and figures as needed and takes full responsibility for the publication’s content, methods, and results.

References

- [1] E. B. Khalmurzaeva та M. O. Orozova, “FELT IN MODERN INTERIOR DESIGN”, *Her. KSUCTA*, №3, 2021, № 3-2021, с. 343–349,. 2021: <https://doi.org/10.35803/1694-5298.2021.3.343-349>.
- [2] I. Burchak та V. Shmelov, “MODERN TRENDS IN RESIDENTIAL INTERIOR DESIGN”, *Theory pract. des.*, № 26, с. 133–139, 2022: <https://doi.org/10.32782/2415-8151.2022.26.16>.
- [3] S. Kim, S. Shah, R. Wunderlich та J. Hasler, “CAD synthesis tools for floating-gate SoC FPAAs”, *Des. Automat. Embedded Syst.*, т. 25, № 3, с. 161–176,. 2021: <https://doi.org/10.1007/s10617-021-09247-9>.
- [4] S. M. Sari, P. F. Nilasari та P. E. D. Tedjokoesoemo, “Implementation of Interior Branding in Retail Interior Design”, *GATR J. Manage. Marketing Rev.*, т. 7, № 1, с. 13–22,. 2022: [https://doi.org/10.35609/jmmr.2022.7.1\(2\)](https://doi.org/10.35609/jmmr.2022.7.1(2)).
- [5] T. Hadjiyanni, “Decolonizing interior design education”, *J. Interior Des.*, т. 45, № 2, с. 3–9, квіт. 2020: <https://doi.org/10.1111/joid.12170>.
- [6] Matsiievskiyi, O.; Honcharenko, T.; Solovei, O.; Liashchenko, T.; Achkasov, I.; Golenkov, V. Using Artificial Intelligence to Convert Code to Another Programming Language. У 2024 IEEE 4th International Conference on Smart Information Systems and Technologies (SIST), Astana, Kazakhstan, 15–17 травня 2024; IEEE, 2024; с 379–385. DOI: 10.1109/sist61555.2024.10629305.
- [7] T. Honcharenko, G. Ryzhakova, Ye. Borodavka, D. Ryzhakov, V. Savenko, O. Polosenko. Method for representing spatial information of topological relations based on a

- multidimensional data model, *ARPN Journal of Engineering and Applied Sciences*, 16(7), 2021, pp. 802–809. https://www.arpnjournals.org/jeas/research_papers/rp_2021/jeas_0421_8555.pdf.
- [8] Y. Riabchun, T. Honcharenko, V. Honta, K. Chupryna, O. Fedusenko. Methods and means of evaluation and development for prospective students' spatial awareness, *International Journal of Innovative Technology and Exploring Engineering*, 8(11), 2019, pp. 4050–4058 DOI:10.35940/ijitee.K1532.0981119.
- [9] Matsiievskiy, O., Achkasov, I., Borodavka, Y., Mazurenko, R. Behavioral model of autonomous robotic systems using reinforcement learning methods (2024) *CEUR Workshop Proceedings*, 3896, pp. 560-568 <https://ceur-ws.org/Vol-3896/short14.pdf>.
- [10] Oleksii Matsiievskiy, Pavlo Kruk, Volodymyr Levytskyi. Neural Network Model for Automating Programming Language Conversion. 2024 *IEEE AITU: Digital Generation Conference* <https://ceur-ws.org/Vol-3966/W3Paper7.pdf>.
- [11] D. Chernyshev, G.Ryzhakova, T. Honcharenko, H. Petrenko, I. Chupryna, N. Reznik. Digital Administration of the Project Based on the Concept of Smart Construction. *Lecture Notes in Networks and Systems*, 495 LNNS, 2023, pp. 1316-1331. DOI: 10.1007/978-3-031-08954-1_114 DOI:10.1007/978-3-031-08954-1_114.
- [12] X. Wu, D. Hong та J. Chanussot, "UIU-Net: U-Net in U-Net for Infrared Small Object Detection", *IEEE Trans. Image Process.*, c. 1, 2022: <https://doi.org/10.1109/tip.2022.3228497>
- [13] B. Li та D. Lima, "Facial expression recognition via ResNet-50", *Int. J. Cogn. Comput. Eng.*, т. 2, с. 57–64., 2021: <https://doi.org/10.1016/j.ijcce.2021.02.002>
- [14] Q. Zhu, "ACDNet with ASPP for Camouflaged Object Detection", *J. Phys.: Conf. Ser.*, т. 1982, № 1, с. 012082., 2021: <https://doi.org/10.1088/1742-6596/1982/1/012082>
- [15] D. Chernyshev, G.Ryzhakova, T. Honcharenko, H. Petrenko, I. Chupryna, N. Reznik, "Digital Administration of the Project Based on the Concept of Smart Construction", *Lecture Notes in Networks and Systems*, 495 LNNS, pp. 1316-1331, 2023. https://doi.org/10.1007/978-3-031-08954-1_114
- [16] A. M. Rostami, M. M. Homayounpour та A. Nickabadi, "Efficient Attention Branch Network with Combined Loss Function for Automatic Speaker Verification Spoof Detection", *Circuits, Syst., Signal Process.*, 2023: <https://doi.org/10.1007/s00034-023-02314-5>