



DOI 10.28925/2663-4023.2026.32.1088

УДК 004.89 (004.49)

Костюк Антон Юрійович

аспірант факультету електронних та інформаційних технологій,
НУ «Чернігівська політехніка», Чернігів, Україна
ORCID: 0009-0001-3107-9976
1999kostyukanton@gmail.com

Зайцев Сергій Васильович

доктор технічних наук, професор кафедри інформаційних та комп'ютерних систем,
НУ «Чернігівська політехніка», Чернігів, Україна
ORCID: 0000-0001-6643-917X
serza1979@gmail.com

Василенко Владислав Михайлович

кандидат технічних наук, т.в.о. відділ інформ. та комунікац. технол.,
ІТГП НАН України
ORCID: 0000-0001-8156-1894
vladvasilenko9@gmail.com

Зайцева Лілія Ігорівна

аспірант Інституту телекомунікацій і глобального інформаційного простору НАН України
ORCID: 0000-0002-0668-711X
lili5990n@ukr.net

РОЗРОБКА ІНТЕРАКТИВНОГО ІНТЕРФЕЙСУ КОРИСТУВАЧА НА ОСНОВІ ДЕТЕРМІНОВАНИХ КІНЦЕВИХ АВТОМАТІВ МУРА

Анотація. У цій статті було розглянуто стандарти якості, а також методики оцінки якості інтерфейсів користувача. Розглянуті методики не підходять для вирішення задачі оцінки якості інтерфейсу користувача через те, що застосовувані методи не дозволяють комплексно оцінити якість інтерфейсу та використовують недостатню кількість показників. Тому для вирішення цього завдання було запропоновано розробити методику оцінки якості інтерактивних інтерфейсів позбавлену виділених недоліків. Зважаючи на це обмеження була запропонована автоматна модель, для проектування інтерактивного інтерфесу. Побудована автоматна модель може бути використана для розв'язання задач проектування інтерфейсу користувача, як звичайних додатків з інтерфейсом користувача під управлінням сучасних ОС, так і для web-додатків. В результаті експерименту встановлено, що розроблені методики проектування інтерактивного інтерфейсу користувача та методика оцінки якості працюють коректно.

Ключові слова: інтерактивні інтерфейси; кінцевий автомат Мура; графічний інтерфейс користувача; користувацькі інтерфейси до баз даних.

ВСТУП

З моменту появи техніки поняття «інтерфейс» стало все частіше з'являтися у нашому житті. Інтерфейс визначається як сукупність можливостей, способів та методів взаємодії двох систем. У випадку взаємодії з машиною/пристроєм використовується поняття інтерфейс користувача, тобто вид інтерфейсу, в якому одна сторона представлена людиною, а інша пристроєм. Інтерактивний інтерфейс – інтерфейс, якому користувач віддає команди машині, та їх виконує та видає результати у зрозумілій для

людини формі, наприклад, візуальними або звуковими сигналами, своєю чергою користувач віддає наступні команди машині використовуючи засоби, що знаходяться в його розпорядженні: кнопками, перемикачами та ін.

За наявності тих чи інших засобів введення, інтерфейси поділяються на типи: жестовий, голосовий, нейрокомп'ютерний і т.д., можливі змішані варіанти. Ці засоби повинні бути необхідними та достатніми, бути зручними та практичними, розташованими та скомпонованими розумно і зрозуміло, відповідати фізіології людини, не повинні призводити до негативних наслідків для організму користувача.

Методами інтерфейсу користувача є набір правил, закладених розробником пристрою, згідно з яким сукупність дій користувача повинна призвести до необхідної реакції пристрою на виконання необхідної задачі – так званий логічний інтерфейс. Ці правила повинні бути досить зрозумілі для розуміння, природні та легкі для запам'ятовування.

Найменший елемент діалогу у літературі, присвяченій проблемам проектування взаємодії в людино-машинних системах зазвичай називають кроком діалогу. Крок діалогу складається з підготовки та видачі повідомлення (запиту) одного з учасників діалогу, а потім підготовки та видачі повідомлення (відповіді) іншого учасника.

Послідовність кроків діалогу називається діалоговою послідовністю. Діалогова послідовність є нормальною, якщо вона реалізується відповідно до діалогової програми без втручання користувача, що змінює «природний перебіг» діалогу. Нормальна діалогова послідовність, що починає і завершує діалог називається повною діалоговою послідовністю. Сукупність всіх можливих повних нормальних діалогових послідовностей, реалізованих діалоговою програмою, є сценарій діалогу.

Постановка проблеми. Поведінка деяких послідовних схем може бути дуже складною, для опису їхньої роботи прості таблиці істинності вже не підходять. Щоб спростити аналіз складних послідовних логічних схем, з'явилася теорія цифрових автоматів для побудови математичних моделей, що використовують абстрактний автомат (Abstract Machine).

Абстрактний автомат – це математична абстрактна модель дискретного пристрою, що має один вхід і один вихід, у кожний момент знаходиться в одному стані з безлічі можливих.

Аналіз останніх досліджень і публікацій. Автомати Мура і Мілі дозволяють описати функціонування цифрових систем, які широко застосовуються при синтезі складних послідовних схем з урахуванням ПЛІС (програмована логічна інтегральна схема). Вони названі на честь своїх винахідників, учених, які розробили теорію автоматів та математичну базу для них у фірмі Bell Labs.

Едвард Форест Мур (1925-2003) опублікував свою першу статтю «Gedankenexperiments on Sequential Machines» у 1956 році.

Джордж Мілі (1927-2010) опублікував «Method of Synthesizing Sequential Circuits» у 1955 році. Згодом він написав першу операційну систему для комп'ютера IBM 704. Пізніше він перейшов працювати в Гарвардський університет.

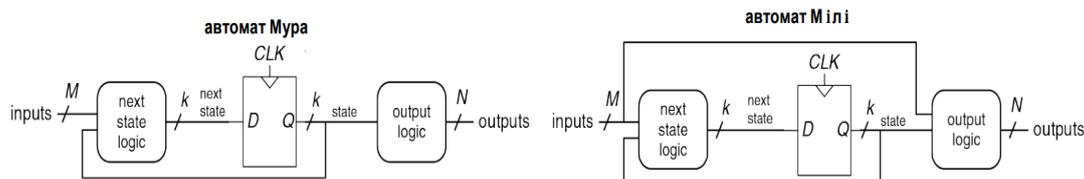


Рисунок 1. Схема порівняння автоматів Мура та Мілі



Схема кінцевого автомата містить пам'ять запам'ятовування допустимих станів. Вихід схеми пам'яті разом із вхідним сигналом надходить на вхідну схему формування переходів, яка формує нове значення в комірці пам'яті (визначає новий стан). Також вихід схеми пам'яті формує вихідний сигнал. Якщо вихідний сигнал повністю визначається значенням осередку пам'яті, то схему називають автомат Мура. У ряді випадків у формуванні вихідного сигналу може задіятися як комірка пам'яті, а й вхідний сигнал. Це дозволяє скоротити кількість необхідних станів. Така схема називається автомат "Мілі".

Основна відмінність між кінцевими автоматами Мура і Мілі у тому, що у автомата Мура зазвичай більше станів, ніж в автоматі Мілі.

Мета статті. Метою статті є розробка та вдосконалення проектування інтерактивного інтерфейсу за допомогою побудови кінцевого автомату Мура. А саме полягає в аналізі, дослідженні та розробці автоматизованого засобу проектування інтерфейсу користувача з модулем оцінки якості.

У рамках цієї роботи були виділені такі завдання:

- дослідження проблем та визначення основних завдань автоматизованої розробки інтерфейсу користувача;
- дослідження принципів та розробка засобів оцінки якості інтерфейсу користувача;
- розробка нотацій для представлення структури та логіки;
- функціонування інтерфейсу користувача.

МЕТОДИКА ДОСЛІДЖЕННЯ

Процес роботи діалогової компоненти (сценарій діалогу) технічної системи може бути представлений графом, який задає переходи від однієї процедури до іншої, тобто зображено діаграмою станів, кожен з яких відображає результат певної фази діяльності користувача. Зазвичай ці фази визначаються таким чином, щоб їм можна було поставити у відповідність інструкції користувача, які можуть бути задані у вигляді команд, комбінацій натискання функціональних клавіш, вибору об'єктів на екрані. Граф задає лише допустимі переходи (можливості користувача), забезпечуючи контроль дій і тим самим стійкість до помилок користувача. Такій схемі відповідає кінцевий детермінований автомат. Він є п'ятіркою наступного виду:

$$A = (X, Y, S, f, j), \quad (1)$$

де $X = \{x \mid i \in 1, \dots, n\}$ – це множина вхідних повідомлень користувача,

$Y = \{y \mid i \in 1, \dots, m\}$ – це множина вихідних повідомлень системи (реплік), $S = \{s \mid i \in 1, \dots, k\}$ – це множина станів автомата, що відповідають стану системи, $f : S \times X \rightarrow S$ – це функція переходів між станами, $j : S \times X \rightarrow Y$ – це функція для виходу.

Якщо розглядати функцію переходів як деяку дію, що виникає в ході реакції на подію, а стан автомата як ту або іншу форму, то роботу програми з інтерактивним інтерфейсом користувача можна описати за допомогою формули (1).

Перебуваючи в певному стані, функція переходу S дозволяє перейти в інший стан, наприклад, змінити вигляд вікна, відкрити нове вікно за цим вхідним символом X .



Представлений кінцевий автомат відповідає автомату Милі, так як функція виходів з автомата залежить і від поточного стану, в якому знаходиться діалогова система, та від вхідного повідомлення, що надійшов від користувача.

Рівність $f(x,s) = s$ означає, що при вхідному повідомленні користувача x автомат, що перебуває у стані s , переходить у стан s , а рівність $j(x,s) = y$ означає, що при цьому на виході з'являється вихідне повідомлення y .

Вимога сюр'єктивності для відображення j не є суттєвим обмеженням. Дійсно, у безліч Y не має сенсу включати елементи, які свідомо що неспроможні з'явитися на виході. Часто безліч виходів визначається «із запасом», тобто більшим, ніж це потрібно насправді. Подання деякого автомата Милі графом відповідає конструкція так званої перехідно-вихідної матриці (або матриці переходів). Це $n \times n$ – матриця виду:

$$M = \| m_{ij} \|, m_{ij} = \{ (s, y) \in X \times Y \mid f(s, x) = s, j(s, x) = y \} \quad (2)$$

Функція виходів для кінцевого детермінованого автомата може бути визначено наступним чином:

$$j : S \rightarrow Y \quad (3)$$

При цьому кінцевий автомат являє собою автомат Мура, в якому функція виходів залежить тільки від поточного стану системи та не залежить від вхідного повідомлення користувача. Автомат Мура може бути представлений як окремий випадок автомата Милі. Якщо у деякого автомата Мура функція виходів відповідає виразу (2), то позначимо $g(s, x) = j(f(s, x))$. Тоді автомат виду $B = (X, Y, S, f, g)$ для всіх непустих вхідних послідовностей формує такі самі вихідні послідовності, як і автомат Мура, якщо, звичайно, не враховувати самий перший вихід автомата.

Повна діалогова послідовність представляється в вигляді деякого шляху на графі сценарію діалогу, що починається з вершини початку діалогу і закінчується в одній із кінцевих вершин. Загалом сценарій діалогу має один початковий крок (початкову вершину у графі) та кілька завершальних кроків (кінцевих вершин у графі) та довільну структуру, тобто в ньому повинні допускатися повернення до попередніх станів та повернення до поточного стану. Такому типу діалогу відповідає кінцевий автомат, що має таку функцію переходів f , що допускається співвідношення такого виду:

$$f(x_k, s_j) = s_j, \quad (4)$$

де $s_j \in S$ та $x_k \in X$.

РЕЗУЛЬТАТИ ДОСЛІДЖЕННЯ

На рисунку 2 показана проста модель взаємодії людини з системою за допомогою інтерактивного інтерфейсу, яка описує кінцеві автомати, які в свою чергу керують входом користувача до системи.

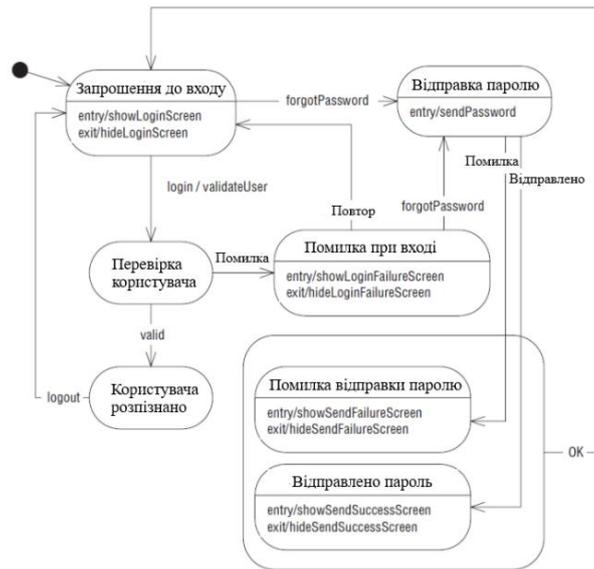


Рисунок 2. Кінцевий автомат, що описує вхід до системи

Скругленими прямокутниками представлені стани. Ім'я стану записується у верхньому відділенні [22]. У нижньому відділенні записуються спеціальні дії, які говорять про те, що потрібно зробити при вході в даний стан та при виході з нього. Наприклад, при вході в стан «Запрошення до входу» викликається дія `showLoginScreen`, та при виході з цього стану – дія `hideLoginScreen`.

Стрілки, що з'єднують стани, називаються переходами. Кожен перехід позначений іменем події, що викликала його, а деякі – також дією, що виконується при спрацьовуванні переходу. Наприклад, якщо, перебуваючи у стані «Запрошення до входу», отримуємо подію `login`, то переходимо в стан «Перевірка користувача» та викликаємо дію `validateUser`. Чорний куток у лівому верхньому кутку діаграми називається початковим псевдостаном. Кінцевий автомат починає роботу переходом із цього псевдостану. Таким чином, на початку роботи наш кінцевий автомат опиняється у стані «Запрошення до входу». Навколо станів «Помилка відправки пароля» та «Пароль успішно відправлено» намальовано суперстан, тому що обидва ці стани реагують на подію «ОК» переходом у стан «Запрошення до входу». Показаний кінцевий автомат наочно пояснює, як працює процедура входу в систему, та розбиває її на невеликі функції. Якщо реалізувати всі функції дій – `showLoginScreen`, `validateUser` і `sendPassword`, – а потім об'єднати їх, дотримуючись представлені на діаграмі логіки, тобто гарантія, що процедура входу працюватиме правильно.

Розглянемо кінцевий автомат. Він керує графічним інтерфейсом тієї частини програми, яка відповідає за вхід у систему. Отримавши початкову подію, автомат виводить вікно входу в систему [21]. Після того, як користувач натисне клавішу `Enter`, автомат перевіряє пароль. Якщо пароль правильний, то автомат переходить у стан `loggedIn` і запускає користувальницький процес (не показаний). В іншому у разі виводиться вікно з повідомленням про помилку введення пароля. Користувач може повторити спробу, натиснувши кнопку `ОК`, або відмовитися від повторних спроб за допомогою кнопки `Скасувати`. Якщо тричі поспіль введено неправильний пароль (подія `thirdBadPassword`), то автомат блокує екран, поки не буде введено пароль адміністратора.

Initial init



```
{
init
{
start loginIn displayLoginScreen
}
loginIn
{
enter checkingPassword checkPassword
cancel init clearScreen
}
checkingPassword
{
passwordGood loggedIn startUserProcess
passwordBad notifyingPasswordBad displayBadPasswordScreen
thirdBadPassword screenLocked displayLockScreen
}
notifyingPasswordBad
{
OK checkingPassword displayLoginScreen
cancel init clearScreen
}
screenLocked
{
enter checkingAdminPassword checkAdminPassword
}
checkingAdminPassword
{
passwordGood init clearScreen
passwordBad screenLocked displayLockScreen
}
}
```

Програма представлена у вигляді кінцевого автомату Мура. Вона описана в одному місці, що зручно для супроводу. Очевидно, що такий підхід застосовується не тільки до графічних інтерфейсів. Його можна використовувати також при побудові текстових та машинних інтерфейсів.

ВИСНОВКИ ТА ПЕРСПЕКТИВИ ПОДАЛЬШИХ ДОСЛІДЖЕНЬ

У статті наведено теоретичний огляд та запропоновано впровадження кінцевого автомату Мура до рішення наукової проблеми оптимізації інтерактивних інтерфейсів шляхом розробки нових моделей та методів. Кінцевий автомат (1) визначає лише внутрішню поведінку системи, не торкаючись поведінки зовнішнього середовища. Для того, щоб система була стійка до аномальних ситуацій довкілля визначимо треба визначити множину E – безліч станів (подій) довкілля. Це може бути переривання від зовнішніх пристроїв або системного програмного забезпечення. У людино-машинних технічних системах вибір здійснюється оператором чи експертом. Введення в контур технічної системи людини, що володіє випадковими та нестаціонарними характеристиками, суттєвими з погляду роботи системи, що ще більше збільшує недетермінованість процесу діалогу. Повністю врахувати поведінку користувача автоматизованої системи стадії проектування діалогу неможливо, але можна виділити деякі найбільш ймовірні кроки користувача, що вимагають негайної реакції системи.



СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Kuang, C. (2020). *User friendly: How the hidden rules of design are changing the way we live, work, and play*. MCD.
2. Wathan, S., & Schoger, S. (2018). *Refactoring UI*.
3. Unger, R., & Chandler, C. (2011). *A project guide to UX design: For user experience designers in the field or in the making*.
4. Li, S., Xu, L. D., & Zhao, S. (2018). 5G internet of things: A survey. *Journal of Industrial Information Integration*, 10, 1–9. <https://doi.org/10.1016/j.jii.2018.01.005>
5. Wroblewski, L. (2011). *Mobile first*. A Book Apart.
6. Cooper, A. (2004). *The inmates are running the asylum: Why high-tech products drive us crazy and how to restore the sanity*. Sams.
7. Krug, S. (2014). *Don't make me think: A common sense approach to web usability* (Revisited ed.). New Riders.
8. Li, K. (2023). Deep learning-based pavement crack detection system. *Journal of Physics: Conference Series*.
9. Hussain, M., & Hanam, R. A. (2024). A comprehensive review of convolutional neural networks for defect detection in industrial applications. *IEEE Access*. <https://ieeexplore.ieee.org/document/10589380>
10. Muin, M., & Abbud, M. (2025). Study of integration of internet of things and deep learning methods for smart city applications. *American Journal of Computer Science and Engineering*, 8(1), 57–68.
11. Harel, J., Koch, C., & Perona, P. (2006). Graph-based visual saliency. In *Advances in neural information processing systems* (NeurIPS). https://proceedings.neurips.cc/paper_files/paper/2006/file/4db0f8b0fc895da263fd77fc8aecabe4-Paper.pdf
12. Achanta, R., Hemami, S., Estrada, F., & Süsstrunk, S. (2009). Frequency-tuned salient region detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (pp. 1597–1604).
13. Fassold, H., & Beiler, W. (2022). Few-shot object detection as a semi-supervised learning problem. In *International Conference on Content-Based Multimedia Indexing*. <https://www.researchgate.net/publication/364245764>
14. Liu, M., & Yao, P. (2024). Robust classification of incomplete time series with noisy labels. In *Proceedings of the 27th International Conference on Computer Supported Cooperative Work in Design (CSCWD)*. <https://www.researchgate.net/publication/382154918>
15. Stephens, B. (2011). Push recovery control for force-controlled humanoid robots. <https://www.researchgate.net/publication/266907694>
16. Pontil, M., & Verri, A. (2001). Support vector machines: Theory and applications. In *Lecture notes in computer science* (pp. 249–257).
17. Mikolov, T., Chen, K., Corrado, G., & Dean, J. (2013). Efficient estimation of word representations in vector space. *arXiv Preprint*. <https://arxiv.org/abs/1301.3781>
18. Simonyan, K., & Zisserman, A. (2015). Very deep convolutional networks for large-scale image recognition. *arXiv Preprint*. <https://arxiv.org/abs/1409.1556>
19. He, K., Zhang, X., Ren, S., & Sun, J. (2016). Deep residual learning for image recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (pp. 770–778).
20. Huang, G., Liu, Z., Van der Maaten, L., & Weinberger, K. Q. (2017). Densely connected convolutional networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.
21. Lin, T.-Y., Maire, M., Belongie, S., et al. (2014). Microsoft COCO: Common objects in context. In *Lecture notes in computer science*.
22. Talasani, H. (2020). Military target detection from satellite images using deep convolutional neural networks. In *Proceedings of the IEEE International Conference on Computing, Communication and Automation*.
23. Zhou, L. (2023). Multi-scale object detector based on coordinate and global information aggregation for UAV aerial images. *Remote Sensing*, 15(14), 3468. <https://www.mdpi.com/2072-4292/15/14/3468>
24. Sabour, S., Frosst, N., & Hinton, G. E. (2017). Dynamic routing between capsules. In *Advances in neural information processing systems* (NeurIPS).

**Anton Kostiuk**

Postgraduate student, Faculty of Electronic and Information Technologies
National University «Chernihiv Polytechnic», Chernihiv, Ukraine
ORCID: 0009-0001-3107-9976
1999kostyukanton@gmail.com

Serhiy Zaitsev

Doctor of Technical Sciences, Professor of the Department of Information and Computer Systems
National University «Chernihiv Polytechnic», Chernihiv, Ukraine
ORCID: 0000-0002-7751-5956
Serza1979@gmail.com

Vladislav Vasylenko

Candidate of Technical Sciences, t.v.o. Department of Information
and Communication Technologies, Institute of Telecommunications
and Global Information Space of the National Academy of Sciences of Ukraine
ORCID: 0000-0001-8156-1894
vladvasilenko9@gmail.com

Lilia Zaitseva

Postgraduate Student, Institute of Telecommunications and Global
Information Space of the National Academy of Sciences of Ukraine
ORCID: 0000-0002-0668-711X
lili5990n@ukr.net

DEVELOPMENT OF THE INTERACTIVE INTERFACE OF KORISTUVACH BASED ON DETERMINATIVE END MOORE AUTOMATICS

Abstract. This article looked at the standards of brilliance, as well as methods for assessing the brilliance of user interfaces. The considered methods are not suitable for the serious task of assessing the quality of the user interface through the fact that the established methods do not allow for a comprehensive assessment of the quality of the interface and compensate for the insufficient number of indicators. Therefore, for this purpose, it was necessary to develop a methodology for assessing the strength of interactive interfaces, eliminating the presence of shortcomings. Importantly, at this point, an automatic model was developed for designing an interactive interface.

An automatic model has been developed that can be used for the development of design tasks for the user interface, both for basic add-ons with the user interface under the control of current operating systems, and for web add-ons.

As a result of the experiment, it was established that the detailed methodology for designing an interactive interface of the computer and the methodology for assessing the quality work correctly.

Keywords: interactive interfaces; Moore's finite state machine; graphical user interface; user interfaces to databases.

REFERENCES (TRANSLATED AND TRANSLITERATED)

1. Kuang, C. (2020). *User friendly: How the hidden rules of design are changing the way we live, work, and play*. MCD.
2. Wathan, A., & Schoger, S. (2018). *Refactoring UI*.
3. Unger, R., & Chandler, C. (2012). *A project guide to UX design: For user experience designers in the field or in the making* (2nd ed.). New Riders.
4. Li, S., Xu, L. D., & Zhao, S. (2018). 5G internet of things: A survey. *Journal of Industrial Information Integration*, 10, 1–9. <https://doi.org/10.1016/j.jii.2018.01.005>
5. Wroblewski, L. (2011). *Mobile first*. A Book Apart.



6. Cooper, A. (2004). *The inmates are running the asylum: Why high-tech products drive us crazy and how to restore the sanity*. Sams.
7. Krug, S. (2014). *Don't make me think: A common sense approach to web usability* (Revisited ed.). New Riders.
8. Li, Q. (2023). Deep learning-based pavement crack detection system. *Journal of Physics: Conference Series*.
9. Hussain, M., & Khanam, R. A. (2024). Comprehensive review of convolutional neural networks for defect detection in industrial applications. *IEEE Access*. <https://ieeexplore.ieee.org/document/10589380>
10. Mueen, M., & Abbood, M. (2025). Investigation of IoT and deep learning techniques integration for smart city applications. *American Journal of Computing and Engineering*, 8(1), 57–68.
11. Harel, J., Koch, C., & Perona, P. (2006). Graph-based visual saliency. In *Advances in neural information processing systems* (NeurIPS). https://proceedings.neurips.cc/paper_files/paper/2006/file/4db0f8b0fc895da263fd77fc8aecabe4-Paper.pdf
12. Achanta, R., Hemami, S., Estrada, F., & Süsstrunk, S. (2009). Frequency-tuned salient region detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (pp. 1597–1604).
13. Fassold, H., & Bailer, W. (2022). Few-shot object detection as a semi-supervised learning problem. In *Proceedings of the International Conference on Content-Based Multimedia Indexing*. <https://www.researchgate.net/publication/364245764>
14. Liu, M., & Yao, P. (2024). Robust classification of incomplete time series with noisy labels. In *Proceedings of the 27th International Conference on Computer Supported Cooperative Work in Design (CSCWD)*. <https://www.researchgate.net/publication/382154918>
15. Stephens, B. (2011). Push recovery control for force-controlled humanoid robots. <https://www.researchgate.net/publication/266907694>
16. Pontil, M., & Verri, A. (2001). Support vector machines: Theory and applications. In *Lecture notes in computer science* (pp. 249–257).
17. Mikolov, T., Chen, K., Corrado, G., & Dean, J. (2013). Efficient estimation of word representations in vector space. *arXiv Preprint*. <https://arxiv.org/abs/1301.3781>
18. Simonyan, K., & Zisserman, A. (2015). Very deep convolutional networks for large-scale image recognition. *arXiv Preprint*. <https://arxiv.org/abs/1409.1556>
19. He, K., Zhang, X., Ren, S., & Sun, J. (2016). Deep residual learning for image recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (pp. 770–778).
20. Huang, G., Liu, Z., Van der Maaten, L., & Weinberger, K. Q. (2017). Densely connected convolutional networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.
21. Lin, T.-Y., Maire, M., Belongie, S., et al. (2014). Microsoft COCO: Common objects in context. In *Lecture notes in computer science*.
22. Talasani, H. (2020). Detection of military targets from satellite images using deep convolutional neural networks. In *Proceedings of the IEEE International Conference on Computing, Communication and Automation*.
23. Zhou, L. (2023). A multi-scale object detector based on coordinate and global information aggregation for UAV aerial images. *Remote Sensing*, 15(14), 3468. <https://www.mdpi.com/2072-4292/15/14/3468>
24. Sabour, S., Frosst, N., & Hinton, G. E. (2017). Dynamic routing between capsules. In *Advances in neural information processing systems* (NeurIPS).

Отримано редакцією журналу / Received: 14.01.26

Прорецензовано / Revised: 30.01.26

Схвалено до друку / Accepted: 26.03.26

