

СУЧАСНІ ПІДХОДИ ДО РОЗПІЗНАВАННЯ ВІЙСЬКОВИХ ОБ’ЄКТІВ ІЗ
ВИКОРИСТАННЯМ МОДЕЛІ YOLOV8

У роботі розглядаються сучасні підходи до розпізнавання військових об’єктів із використанням нейромережевої моделі YOLOv8. Основну увагу приділено модифікаціям nano та small, які забезпечують оптимальний баланс між швидкодією та точністю. Навчання здійснюється протягом 50 епох на основі підготовленого набору даних, що містить 12 класів різноманітних військових і цивільних об’єктів. Проведено розмітку зображень та аналіз ефективності моделей в умовах наближених до реальних бойових сценаріїв. Отримані результати свідчать про високу придатність легких версій YOLOv8 для виконання оборонних і розвідувальних завдань у режимі реального часу. Окрему увагу приділено можливості інтеграції навчених моделей у безпілотні літальні апарати (БпЛА) для автоматичного виявлення цілей.

Ключові слова: YOLOv8, нейронні мережі, Python, набір даних, розмітка, військові об’єкти.

У контексті зростання масштабів військових конфліктів актуальним завданням є впровадження сучасних технологій в оборонну та розвідувальну діяльність. Одним із перспективних напрямів є використання систем комп’ютерного зору на базі глибокого навчання для автоматизованого виявлення об’єктів. У межах даного дослідження обрано підхід із використанням моделі YOLOv8 – одного із нових поколінь алгоритмів виявлення об’єктів, розробленого компанією Ultralytics.

YOLOv8 відзначається підвищеною точністю та швидкодією порівняно з попередніми версіями, а також підтримкою уніфікованої архітектури для трьох основних задач комп’ютерного зору: виявлення об’єктів, сегментації екземплярів і класифікації зображень. Моделі цього сімейства варіюються за розміром і продуктивністю – від легкої та швидкої YOLOv8n (nano) до потужної, проте повільнішої YOLOv8x (extra large).

З урахуванням потреб реального часу та обмежень обчислювальних ресурсів для експериментів було обрано варіанти YOLOv8n та YOLOv8s, які поєднують високу швидкість роботи з прийнятною точністю.

Таблиця 1

Порівняння моделей YOLOv8 nano та small [1]

Model	size (pixels)	mAPval 50-95	Speed CPU ONNX (ms)	Speed A100 TensorRT (ms)	params (M)	FLOPs (B)
YOLOv8n	640	37.3	80.4	0.99	3.2	8.7
YOLOv8s	640	44.9	128.4	1.20	11.2	28.6

Після визначення архітектури моделі та її конфігурації було обрано відкритий датасет для навчання – Military Assets Dataset (12 Classes – YOLOv8 Format) <https://www.kaggle.com/datasets/raws118/military-assets-dataset-12-classes-yolo8-format>, розміщений на платформі Kaggle. Набір призначений для задач виявлення та класифікації об’єктів у військовому середовищі. Він містить 26 315 розмічених зображень, розподілених на 12 класів та поділених на підмножини:

- навчальна вибірка: 21 978 зображень;
 - валідаційна вибірка: 2 941 зображення;
 - тестова вибірка: 1 396 зображень.
- Список класів:
- camouflage_soldier – солдати в маскувальному спорядженні;
 - weapon – стрілецька та інша зброя;
 - military_tank – бронетехніка з важким озброєнням;
 - military_truck – вантажні військові автомобілі;
 - military_vehicle – інші військові транспортні засоби;
 - civilian – неозброєні цивільні особи;
 - soldier – військовослужбовці без камуфляжу;

- civilian_vehicle – цивільні автомобілі;
- military_artillery – великокаліберні артилерійські системи;
- trench – бойові оборонні укріплення;
- military_aircraft – літаки та гелікоптери військового призначення;
- military_warship – бойові кораблі.

Цей набір даних є репрезентативною основою для навчання моделей штучного інтелекту, орієнтованих на автоматизоване виявлення загроз, аналіз ситуацій та підтримку розвідувальних операцій у режимі реального часу.

Після завантаження набору даних було виконано статистичний аналіз міток (рис. 1). На діаграмі розподілу класів (верхній лівий графік) чітко видно сильний дисбаланс: клас military_tank домінує з понад 17 000 зображень, тоді як класи civilian, civilian_vehicle та military_artillery значно непередставлені – менш ніж 1 000 зразків кожен.

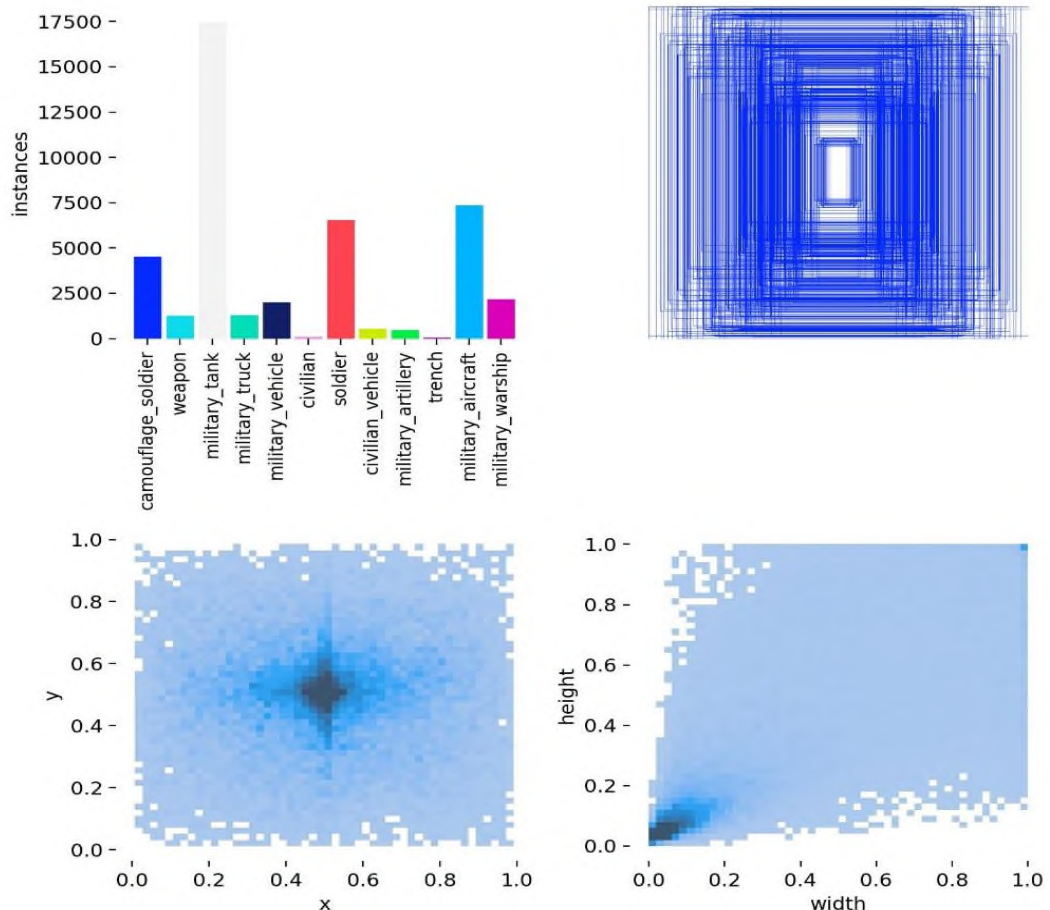


Рис. 1. Інформація про набір даних Military

Джерело: сформовано автором

Таке співвідношення може спричинити перебіс у навчанні моделі, зниження точності виявлення рідкісних класів та погіршення генералізації.

На графіку в правому верхньому куті візуалізовано накладання всіх об'єктів у наборі даних, що демонструє велику кількість варіантів розташування і розмірів боксів – однак переважають центрально розміщені об'єкти зі стандартними пропорціями.

Два нижні графіки на рис. 1 деталізують просторові характеристики боксів. Графік x/y-координат центру об'єктів свідчить про високу щільність об'єктів поблизу центру зображення ($x \approx 0.5$, $y \approx 0.5$) [2]. Графік ширини/висоти bounding box'ів вказує, що більшість виявлених об'єктів є відносно компактними, з шириною та висотою в межах 0.1-0.2 нормалізованих одиниць.

Під час навчання необхідно враховувати дисбаланс класів та геометричних характеристик об'єктів під час навчання моделей YOLOv8, особливо при виборі функцій втрат і аугментації.

На рис. 2 наведено вміст набору даних з лейблами по закінченню процесу навчання, де номер – це об'єкт класу.

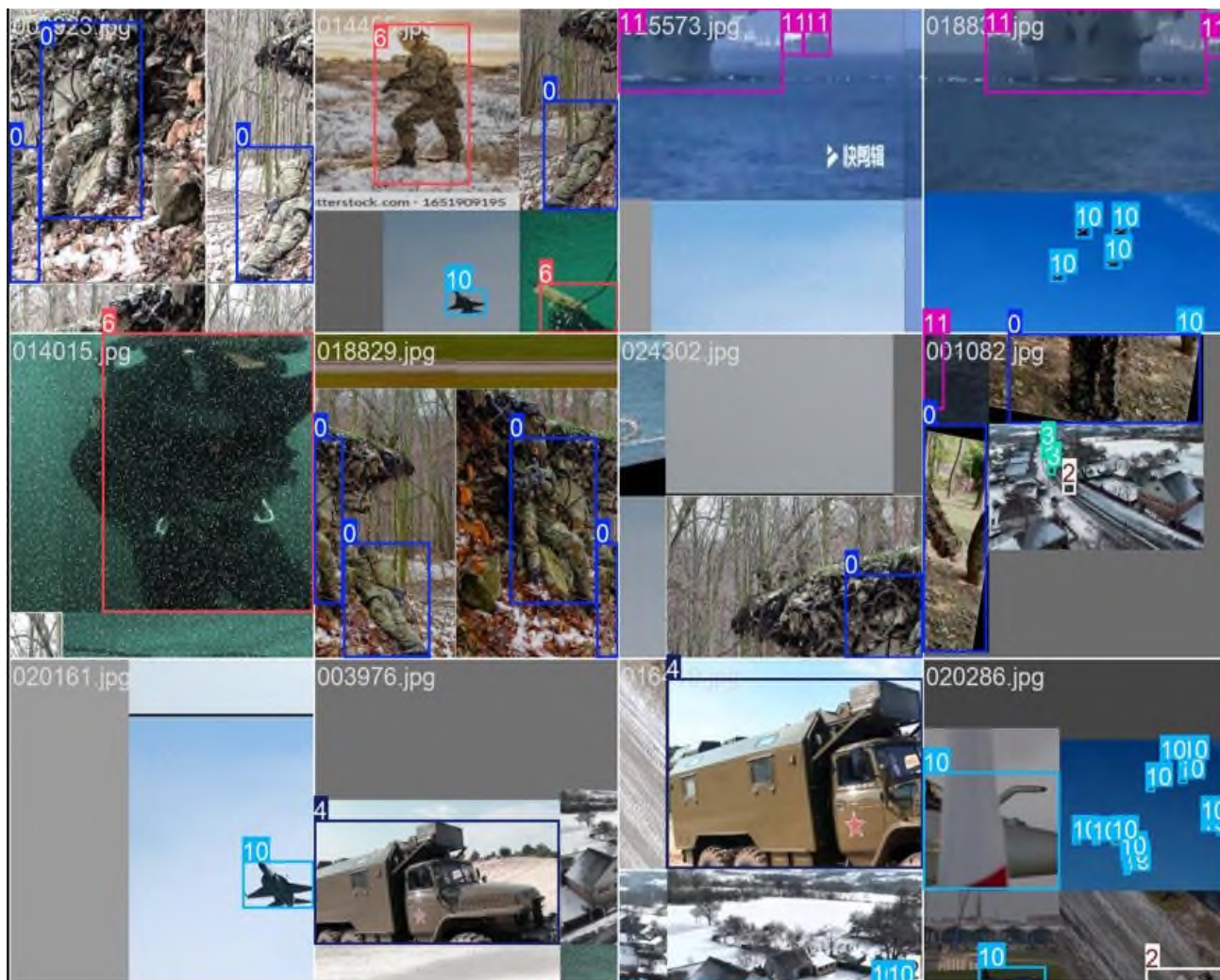


Рис 2. Вміст набору даних з лейблами
Джерело: сформовано автором

Першим кроком стала побудова базової моделі на основі архітектури YOLOv8 nano (YOLOv8n), однієї з найлегших модифікацій, оптимізованих для високої швидкодії з мінімальними вимогами до ресурсів. Навчання моделі здійснювалося на підготовленому наборі даних із 12 класами впродовж 50 епох, що є достатнім для базової первинної конвергенції без перенавчання (рис. 3).

Як оптимізатор використовувався SGD (Stochastic Gradient Descent) зі стандартними параметрами за замовчуванням, передбаченими реалізацією в YOLOv8 [3]. Такий вибір забезпечує стабільне оновлення вагів, хоча й може поступатися адаптивним методам (типу Adam) за швидкістю збіжності на складніших задачах.

Під час навчання здійснювався моніторинг метрик точності (precision, recall, mAP), а також втрат (losses) по класах та по об'єктах, що дозволило сформулювати базову оцінку ефективності моделі та виявити потенційні проблеми, пов'язані з дисбалансом даних.

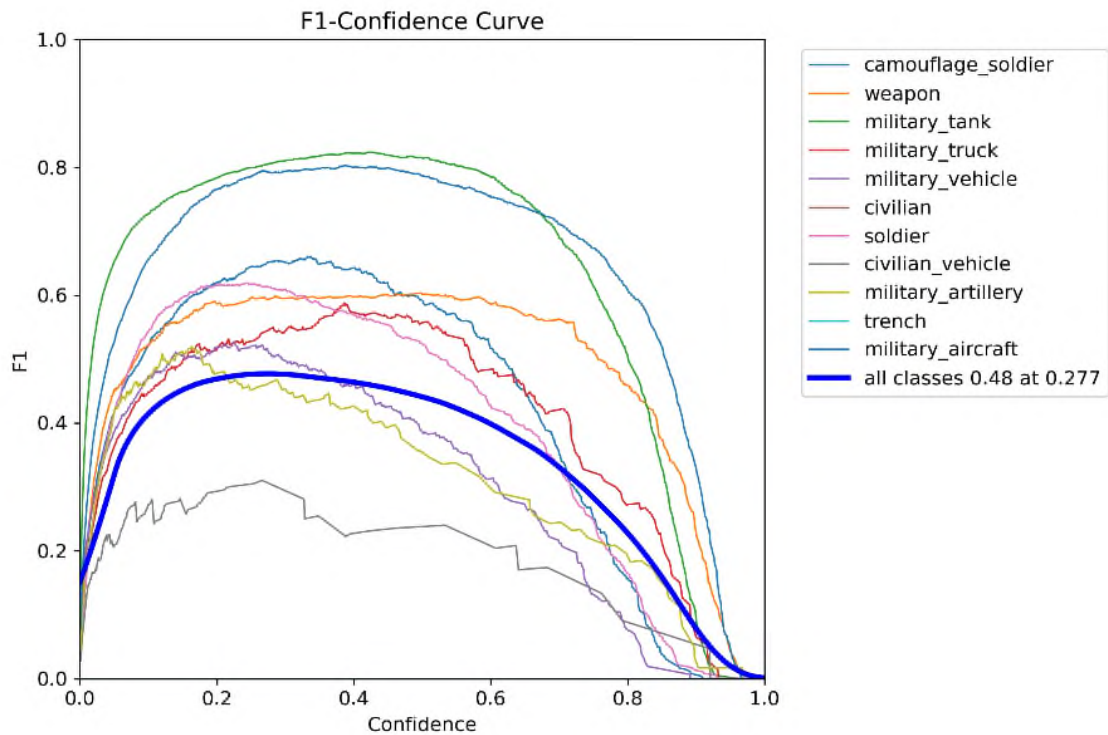


Рис. 3. Результат навчання YOLOv8 nano, 50 епох
Джерело: сформовано автором

F1-Confidence Curve демонструє як змінюється F1-метрика (гармонічне середнє точності та повноти) залежно від порогу впевненості (confidence threshold) [4] моделі для кожного з 12 класів. Найкращі класифікаційні результати модель досягла на класах military_tank, camouflage_soldier та military_aircraft. Слабка ефективність на класах civilian_vehicle і military_artillery – через недостатню кількість зображень у цих класах. Ідеальний поріг для усіх класів – близько 0.28, при якому модель досягає глобального максимуму F1. Загальна F1-метрика 0.48 свідчить про середню якість розпізнавання: модель виявляє об'єкти, але помилки залишаються значними. Цього недостатньо для високоточної автоматизованої системи, особливо в критичних сценаріях (наприклад, у бойових умовах).

Для порівняння було обрано модель YOLOv8 small з аналогічною тривалістю навчання у 50 епох (рис. 4). У результаті F1-score знизився до 0.36 при оптимальному порозі довіри 0.298, що свідчить про деяке погіршення якості розпізнавання об'єктів порівняно з моделлю YOLOv8 nano.

Таке зниження продуктивності вказує на важливість збалансованості класів у наборі даних. Наявність переважання одного класу над іншими, зокрема military_tank, значно впливає на здатність моделі навчатися розпізнаванню менш представлених об'єктів. Окрім цього, збереження сталої кількості епох у поєднанні з використанням оптимізатора SGD за замовчуванням може обмежувати потенціал моделі YOLOv8 small, яка має більшу кількість параметрів і потребує більш гнучких стратегій навчання. Підключення іншого оптимізатора, наприклад AdamW, а також збільшення кількості епох потенційно дозволило б досягти кращих результатів.

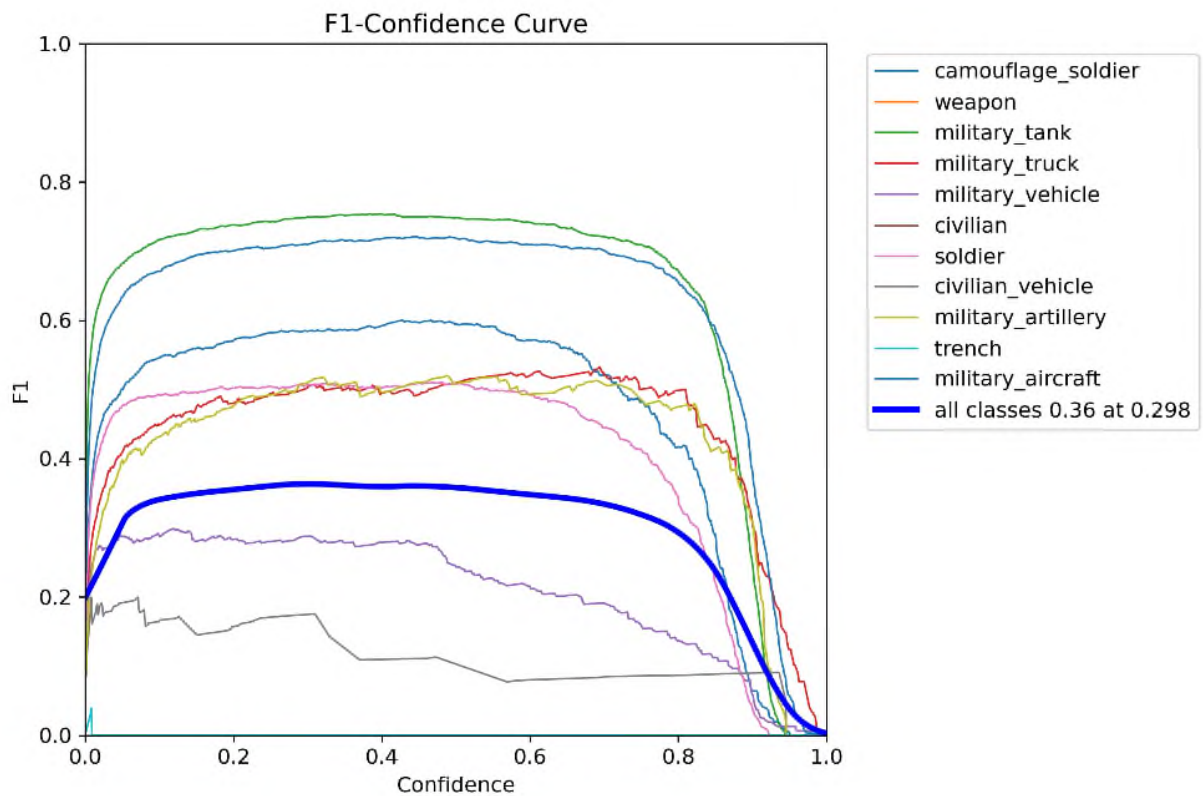


Рис.4. Результат навчання YOLOv8 small, 50 епох
Джерело: сформовано автором

Результат роботи YOLOv8 папо на тестових даних наведено на рис. 5.



Рис. 5. Результат роботи YOLOv8 папо на тестових даних
Джерело: сформовано автором

Втім, незважаючи на зниження точності, модель залишається придатною для практичних експериментів. Зокрема, її можна успішно застосовувати у тестових вильотах безпілотних літальних апаратів, де ключовим є не лише абсолютна точність, а й стабільність роботи та здатність працювати в умовах реального часу. Таким чином, навіть із поточними показниками, YOLOv8 папо демонструє перспективність для створення прототипу та подальшого вдосконалення в рамках розробки систем реального часу для виявлення військових об'єктів.

References

1. Alawi, M., & Mohammed, A. (2024). The Role of YOLOv8 in Enhancing Strategic Military Equipment Detection. ResearchGate.
2. Singh, S., & Ratna, G. N. (2024). Military Based Object Detection in Satellite Imagery by Optimising YOLOv8. IEEE Space, Aerospace and Defence Conference.
3. Alawi, M., & Mohammed, A. (2024). Empowering Military Vehicle Detection and Classification with YOLOv8 Model. ResearchGate.
4. Zhao, B., Zhou, Y., Song, R. et al. (2024) Modular YOLOv8 optimization for real-time UAV maritime rescue object detection. Sci Rep 14, 24492.

DOI 10.34132/mspc2025.01.14.16

**Maksym Salyutin,
Ievgen Sidenko,
Yuriy Kondratenko,**

MODERN APPROACHES TO MILITARY OBJECT RECOGNITION USING THE YOLOV8 MODEL

The paper discusses modern approaches to military object recognition using the YOLOv8 neural network model. The main attention is paid to nano and small modifications, which provide an optimal balance between speed and accuracy. Training is carried out over 50 epochs based on a prepared dataset containing 12 classes of various military and civilian objects. The images were labeled and the models' performance was analyzed under conditions close to real combat scenarios. The results obtained indicate the high suitability of lightweight versions of YOLOv8 for real-time defense and reconnaissance tasks. Special attention is paid to the possibility of integrating trained models into unmanned aerial vehicles (UAVs) for automatic target detection.

Keywords: YOLOv8, neural networks, Python, dataset, markup, military objects.

Відомості про автора / Information about the Author

Максим Салютін, аспірант кафедри інтелектуальних інформаційних систем, Чорноморського національного університету імені Петра Могили, м. Миколаїв, Україна. E-mail: salyutin1@gmail.com, orcid: <https://orcid.org/0009-0003-0788-0012>.

Maksym Salyutin, PhD student of the Intelligent Information Systems Department of the Petro Mohyla Black Sea National University, Mykolaiv, Ukraine. E-mail: salyutin1@gmail.com, orcid: <https://orcid.org/0009-0003-0788-0012>.

Євген Сіденко, канд. техн. наук, доцент кафедри інтелектуальних інформаційних систем, Чорноморського національного університету імені Петра Могили, м. Миколаїв, Україна. E-mail: ievgen.sidenko@chmnu.edu.ua, orcid: <https://orcid.org/0000-0001-6496-2469>.

Ievgen Sidenko, PhD, Associate Professor of the Intelligent Information Systems Department of the Petro Mohyla Black Sea National University, Mykolaiv, Ukraine. E-mail: ievgen.sidenko@chmnu.edu.ua, orcid: <https://orcid.org/0000-0001-6496-2469>.

Юрій Кондратенко, д-р техн. наук, професор, завідувач кафедри інтелектуальних інформаційних систем, Чорноморського національного університету імені Петра Могили, м. Миколаїв, Україна. E-mail: yuriy.kondratenko@chmnu.edu.ua, orcid: <https://orcid.org/0000-0001-7736-883X>.

Yuriy Kondratenko, Doctor of Science, Professor, Head of the Intelligent Information Systems Department at Petro Mohyla Black Sea National University, Mykolaiv, Ukraine. E-mail: yuriy.kondratenko@chmnu.edu.ua, orcid: <https://orcid.org/0000-0001-7736-883X>.