

КВАЛІФІКАЦІЙНА РОБОТА

на здобуття освітнього ступеня

бакалавр

(назва освітнього ступеня)

на тему: Застосування машинного навчання для розпізнавання об'єктів на зображеннях

Виконав: студент IV курсу, групи СН-41

спеціальності 122 Комп'ютерні науки

(шифр і назва спеціальності)

(підпис)

Іщук В.Б.

(прізвище та ініціали)

Керівник

(підпис)

Марценко С.В.

(прізвище та ініціали)

Нормоконтроль

(підпис)

Шимчук Г.В.

(прізвище та ініціали)

Завідувач кафедри

(підпис)

Боднарчук І.О.

(прізвище та ініціали)

Рецензент

(підпис)

Паламар А.М.

(прізвище та ініціали)

Міністерство освіти і науки України
Тернопільський національний технічний університет імені Івана Пулюя

Факультет комп'ютерно-інформаційних систем і програмної інженерії
(повна назва факультету)

Кафедра комп'ютерних наук
(повна назва кафедри)

ЗАТВЕРДЖУЮ

Завідувач кафедри

Боднарчук І.О.
(підпис) (прізвище та ініціали)

« 21 » червня 2025 р.

**ЗАВДАННЯ
НА КВАЛІФІКАЦІЙНУ РОБОТУ**

на здобуття освітнього ступеня Бакалавр
(назва освітнього ступеня)

за спеціальністю 122 Комп'ютерні науки
(шифр і назва спеціальності)

Студенту Іщук Валентин Богданович
(прізвище, ім'я, по батькові)

1. Тема роботи Застосування машинного навчання для розпізнавання об'єктів на зображеннях

Керівник роботи Марценко Сергій Володимирович, к.т.н., доц., доцент кафедри КН
(прізвище, ім'я, по батькові, науковий ступінь, вчене звання)

Затверджені наказом ректора від « 7 » травня 2025 року № 4/7-444

2. Термін подання студентом завершеної роботи 21.06.2025р.

3. Вихідні дані до роботи Зображення з певними об'єктами які потрібно розпізнати розміщена на сайті

4. Зміст роботи (перелік питань, які потрібно розробити)

Вступ. РОЗДІЛ 1. Розгляд типових рішень та засобів для розпізнавання об'єктів на зображеннях 1.1 Розпізнавання об'єктів на зображеннях 1.2 Типові підходи до розпізнавання об'єктів на зображеннях 1.3 Перший етап збір та підготовка даних 1.4 Другий етап попереднє опрацювання даних 1.5 Виділення ознак (feature extraction) 1.7 Розпізнавання на нових даних або Inference 1.7 Етап оцінки якості розпізнавання. Використання метрик точності 1.8 Висновок до першого розділу.

РОЗДІЛ 2. Розробка програми для розпізнавання об'єктів на зображенні. 2.1 Переваги та аналіз версії нейронної мережі YOLO v8 2.2 Попередня обробка 2.3 Застосування карти ознак (feature maps) 2.4 Зображення розпізнаних об'єктів (Inference) 2.5 Розпізнавання об'єктів на зображенні та виділення їх контуром 2.6 Висновок до другого розділу

РОЗДІЛ 3. Безпека життєдіяльності, основи охорони праці. 3.1 Безпека життєдіяльності.

Мета та завдання. 3.2 Інформаційне забезпечення БЖД. 3.3 План ліквідації аварій на виробничому об'єкті. 3.4 Висновок до третього розділу. Висновки. Перелік джерел

5. Перелік графічного матеріалу (з точним зазначенням обов'язкових креслень, слайдів)

Слайди: Тема 1. Мета та задачі 2. Актуальність роботи 3. Порівняння типових методів для розпізнавання 4. Порівняння методів для розпізнавання більш детально 5. Основні етапи розпізнавання об'єктів на зображенні 6. Розпізнавання об'єктів на зображенні згідно етапів обробки (yolo v8n) 7. Розпізнавання об'єктів на зображенні згідно етапів обробки (yolo v8m) 8. Розпізнавання об'єктів на зображенні згідно етапів обробки (Yolo v8x) 9. Висновки 10.

6. Консультанти розділів роботи

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв
Безпека життєдіяльності, основи охорони праці	к.т.н., доц. Сенчишин Віктор Степанович	12.06.2025	15.06.2025

7. Дата видачі завдання 29 січня 2025 р.
КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів роботи	Термін виконання етапів роботи	Примітка
1.	Ознайомлення з завданням до кваліфікаційної роботи	30.01.2025	
2.	Підбір джерел необхідних для розробки програми	31.01.2025-03.02.2025	
3.	Опрацювання джерел по темі кваліфікаційної роботи	04.02.2025-06.02.2025	
4.	Виконання дослідження щодо розробки програми	07.02.2025-11.02.2025	
	Розроблення програми		
5.	Оформлення розділу «Розгляд типових рішень та засобів для розпізнавання об’єктів на зображеннях»	03.06.2025-05.06.2025	
6.	Оформлення розділу «Розробка програми для розпізнавання об’єктів на зображенні»	06.06.2025-08.06.2025	
7.	Виконання завдання до підрозділу «Безпека життєдіяльності»	09.06.2025-11.06.2025	
8.	Виконання завдання до підрозділу «Основи охорони праці»	12.06.2025-13.06.2025	
9.	Оформлення кваліфікаційної роботи	14.06.2025-15.06.2025	
10.	Нормоконтроль	16.06.2025-19.06.2025	
11.	Перевірка на плагіат	20.06.2025	
12.	Попередній захист кваліфікаційної роботи	21.06.2025	
13.	Захист кваліфікаційної роботи	25.06.2025	

Студент

(підпис)

Іщук В.Б

(прізвище та ініціали)

Керівник роботи

(підпис)

Марценко С.В.

(прізвище та ініціали)

АНОТАЦІЯ

Застосування машинного навчання для розпізнавання об'єктів на зображеннях
// Кваліфікаційна робота освітнього рівня «Бакалавр» // Іщук Валентин
Богданович // Тернопільський національний технічний університет імені Івана
Пулюя, факультет комп'ютерно-інформаційних систем і програмної інженерії,
кафедра комп'ютерних наук, група СН-41 // Тернопіль, 2025 // С. , рис. – , табл.
– , кресл. – , додат. – , бібліогр. – .

Ключові слова: зображення, об'єкт, розпізнавання зображення.

Кваліфікаційна робота присв'ячена розробці програми по розпізнаванню об'єктів на зображенні, для прикладу взято в якості об'єктів на зображення - кота та людину. В роботі здійснена розробка програми по розпізнаванню об'єктів при цьому проаналізовано технології які використовуються під час розпізнавання. Описано основні поняття, принципи та етапи які необхідні для розпізнавання об'єктів на зображенні. При цьому використані різні моделі нейронних мереж (Yolo v8n, Yolo v8m, Yolo v8x) та здійснено їх порівняння результатів виявлення об'єктів на зображенні.

В першому розділі кваліфікаційної роботи описано типи технологій, які застосовуються при розпізнаванні об'єктів на зображенні. Наведений опис всіх етапів розпізнавання об'єктів на зображенні та дана їх коротка характеристика. У другому розділі описано розробку програми для розпізнавання об'єктів, для прикладу ми обрали в якості об'єктів – людину та кота. Проведено порівняльний аналіз різних моделей нейронних мереж (Yolo v8n, Yolo v8m, Yolo v8x) та обрана та яка найкраще здійснює розпізнавання об'єктів на зображенні.

У третьому розділі кваліфікаційної роботи було опрацьовано такі питання як поняття мети та завдання БЖД це стосувалося безпеки життєдіяльності. Крім цього розкриті питання при дії при ліквідації аварій на об'єкті. А також наведена характеристика катастроф і типові дії та аварійно-відновні роботи при ліквідації наслідків катастроф та аварій.

Об'єкт дослідження: Процес розпізнавання об'єктів на зображенні.

Предмет дослідження: Технології та методи які застосовуються при розпізнаванні об'єктів на зображенні.

Практичне значення одержаних результатів полягає у створенні програми для розпізнавання об'єктів на зображенні, яка може бути після доопрацювання використана в системах, наприклад, відеоспостереження.

ANNOTATION

Application of machine learning for object recognition in images // Qualification work of the educational level “Bachelor” // Ishchuk Valentyn Bogdanovych // Ivan Pulyuy Ternopil National Technical University, Faculty of Computer Information Systems and Software Engineering, Department of Computer Science, Group SN-41 // Ternopil, 2025 // P. , fig. – , tab. – , drawing – , append. – , bibliography – .

Keywords: image, object, image recognition.

The qualification work is devoted to the development of a program for recognizing objects in an image, for example, a person and a person are taken as objects in an image. The work develops programs for recognizing objects using this analyzed technology, which is used during recognition. The basic concepts, principles and stages necessary for recognizing objects in an image are described. In this case, different models of neural networks (Yolo v8n, Yolo v8m, Yolo v8x) are used and their results for objects in an image are compared. The first section of the qualification work describes the types of technologies used in recognizing objects in an image. A description of all stages of recognizing objects in an image is given and their brief characteristics are given. The second section describes the development of a program for recognizing objects, for example, we chose a person and a cat as objects. A comparative analysis of various neural network models (Yolo v8n, Yolo v8m, Yolo v8x) was conducted and the best recognition of objects in the image was selected.

In the third section of the qualification work, such issues as the concept of the purpose and task of the BZH were worked out, this concerned the safety of life. In addition, the issue of actions during the elimination of accidents at the facility was revealed. And the characteristics of disasters and types of actions and emergency recovery work during the elimination of the consequences of disasters and accidents are also given.

Object of study: The process of recognizing objects in the image.

Subject of study: Technologies and methods used in recognizing objects in the image.

The practical significance of the results obtained was in the creation of a program for recognizing objects in the image, which can be used in systems, for example, video surveillance, after refinement.

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ, СКОРОЧЕНЬ І ТЕРМІНІВ

БЖД – Безпека життєдіяльності.

CNN – Convolutional Neural Networks.

IoU – Intersection over Union - для локалізації.

KNN – k-Nearest Neighbors.

mAP – mean Average Precision - для комплексної оцінки.

SVM – Support Vector Machine.

ЗМІСТ

ВСТУП	8
РОЗДІЛ 1. РОЗГЛЯД ТИПОВИХ РІШЕНЬ ТА ЗАСОБІВ ДЛЯ РОЗПІЗНАВАННЯ ОБ'ЄКТІВ НА ЗОБРАЖЕННЯХ	9
1.1 Розпізнавання об'єктів на зображеннях	9
1.2 Типові підходи до розпізнавання об'єктів на зображеннях.....	10
1.3 Перший етап збір та підготовка даних.....	11
1.4 Другий етап попереднє опрацювання даних	13
1.5 Третій етап виділення ознак (feature extraction).....	14
1.7 Четвертий етап розпізнавання на нових даних або Inference	18
1.7 Етап оцінки якості розпізнавання. Використання метрик точності...	19
1.8 Висновок до першого розділу	29
РОЗДІЛ 2. РОЗРОБКА ПРОГРАМИ ДЛЯ РОЗПІЗНАВАННЯ ОБ'ЄКТІВ НА ЗОБРАЖЕННІ	30
2.1 Переваги та аналіз версії нейронної мережі YOLO v8.....	30
2.2 Попередня обробка.....	33
2.3 Застосування карти ознак (feature maps).....	38
2.4 Зображення розпізнаних об'єктів (Inference)	39
2.5 Розпізнавання об'єктів на зображенні та виділення їх контуром	41
2.6 Висновок до другого розділу	43
РОЗДІЛ 3. БЕЗПЕКА ЖИТТЄДІЯЛЬНОСТІ, ОСНОВИ ОХОРОНИ ПРАЦІ	44
3.1 Безпека життєдіяльності. Мета та завдання	44
3.1 Інформаційне забезпечення БЖД	46
3.1 План ліквідації аварій на виробничому об'єкті	48
3.1 Висновок до третього розділу	53
ВИСНОВКИ.....	54
ПЕРЕЛІК ДЖЕРЕЛ	55
ДОДАТКИ	

ВСТУП

Актуальність теми. Розпізнавання об'єктів на зображеннях є ключовим напрямом комп'ютерного зору та штучного інтелекту, який знаходить широке застосування у різних галузях - від медицини, безпеки, автономного транспорту до промислового контролю якості та розумних міст. Зі зростанням обсягів візуальних даних зростає потреба в автоматизованих системах, здатних ефективно і точно ідентифікувати об'єкти в складних умовах - при різній освітленості, масштабі, фоні та перекриттях. Сучасні алгоритми глибокого навчання дозволяють досягати високої точності розпізнавання, забезпечуючи нові можливості для автоматизації процесів, підвищення безпеки та прийняття рішень на основі зображень. Тому створення програм на основі машинного навчання для розпізнавання об'єктів на зображеннях є важливим та актуальним завданням.

Мета і задачі дослідження. Метою даної кваліфікаційної роботи освітнього рівня «Бакалавр» є огляд методів машинного навчання та розробка програми яка дозволяє розпізнавати об'єкти на зображеннях. Для досягнення поставленої мети потрібно виконати ряд завдань, зокрема:

1. Провести аналіз типових етапів розпізнавання об'єктів на зображеннях;
2. Провести аналіз методів машинного навчання;
3. Здійснити розробку програми для розпізнавання об'єктів на зображенні;
4. Здійснити порівняння роботи методів розпізнавання об'єктів на зображеннях;
5. Провести тестування розробленої програми.

Практичне значення одержаних результатів.

Даний програмний продукт може бути використано на практиці для задач розпізнавання об'єктів на зображенні зокрема, котів та людей. Тому створення програми має практичне значення для задач розпізнавання об'єктів на зображенні і може бути використано в системах відеоспостереження.

РОЗДІЛ 1. РОЗГЛЯД ТИПОВИХ РІШЕНЬ ТА ЗАСОБІВ ДЛЯ РОЗПІЗНАВАННЯ ОБ'ЄКТІВ НА ЗОБРАЖЕННЯХ

1.1 Розпізнавання об'єктів на зображеннях

Перед описом розробки розглянемо спочатку де застосовується розпізнавання об'єктів на зображеннях. Необхідно сказати що це одна з найважливіших і найпопулярніших задач комп'ютерного зору. Ось кілька основних сфер, де це використовується на практиці:

1. Безпека та відеоспостереження. Автоматичне виявлення людей, транспорту, підозрілих предметів. Розпізнавання облич для контролю доступу. Аналіз поведінки (наприклад, виявлення натовпу, агресії).

2. Автомобільна промисловість (Автономні авто). Виявлення пішоходів, інших автомобілів, дорожніх знаків і перешкод для безпечного руху. Підтримка систем допомоги водієві (ADAS).

3. Медична діагностика. Автоматичне розпізнавання патологій на медичних знімках (рентген, МРТ, УЗД). Підрахунок клітин або виявлення аномалій.

4. Роздрібна торгівля та маркетинг. Аналіз поведінки покупців (кількість, час перебування біля товарів). Автоматичний облік товарів, розпізнавання продукції на полицях.

5. Виробництво та промисловість. Контроль якості продукції (виявлення дефектів). Автоматизація складання і упаковки.

6. Робототехніка. Орієнтація в просторі, виявлення і розпізнавання об'єктів для взаємодії. Навігація і маніпулювання.

7. Допоміжні технології. Допомога людям з вадами зору (опис об'єктів, навігація). Переклад текстів та об'єктів у реальному часі.

8. Соціальні мережі та розваги. Фільтри на обличчя, доповнена реальність (AR). Категоризація і пошук зображень.

Як ми бачимо область використання задач розпізнавання об'єктів на зображеннях доволі широка тому саме ці задачі є актуальними.

1.2 Типові підходи до розпізнавання об'єктів на зображеннях

Розпізнавання об'єктів на зображеннях зазвичай складається з кількох основних етапів. Розглянемо спочатку ці етапи.

1. Збір та підготовка даних. На цьому етапі можуть бути використані:

- Збір зображень для навчання (dataset).
- Анотування (розмітка) об'єктів: визначення bounding boxes, класів об'єктів тощо.

- Очистка, балансування класів, збільшення (augmentation) даних для покращення якості навчання.

2. Наступним етапом є попередня обробка.

- Масштабування, нормалізація розмірів зображень.
- Усунення шуму.
- Зміна кольорової гами (наприклад, в RGB чи grayscale).
- Балансування яскравості, контрастності.

3. Третім етапом є виділення ознак (feature extraction):

При цьому у класичних підходах, наприклад, SIFT, HOG; у сучасних системах цей етап автоматично виконується нейронними мережами.

- Пошук характерних рис зображення, які допомагають ідентифікувати об'єкт.

- Побудова дескрипторів об'єктів.

4. Наступним етапом є навчання моделі та її використання для розпізнавання:

- Тут застосовуються класичні методи: SVM, KNN, Decision Trees.
- Сучасні методи: глибокі нейронні мережі (CNN - Convolutional Neural Networks), YOLO, Faster R-CNN, SSD.
- Підбір гіперпараметрів, оптимізація моделі.

5. Наступним кроком має бути розпізнавання (Inference):

На цьому етапі відбувається застосування натренованої моделі до нових зображень. При цьому відбувається визначення:

- Класу об'єкта,
- Координат об'єкта на зображенні (локалізація),
- Ймовірності належності до певного класу.

6. Наступний етап це оцінка якості розпізнавання. Використання метрик точності таких як:

- Accuracy, Precision, Recall, F1-score — для класифікації.
- IoU (Intersection over Union) — для локалізації.
- mAP (mean Average Precision) — для комплексної оцінки.

7. Аналіз помилок, донавчання моделі.

Після цього можна займатись інтеграцією отриманої прогами в діючі системи наприклад система відеоспостереження чи інші. Коротко розглянемо ці етапи.

1.3 Перший етап збір та підготовка даних

На цьому етапі ми маємо зібрати якомога більше зображень, які відповідають майбутній задачі. Наприклад, якщо хочемо навчити модель розпізнавати:

- людей - збираємо фото людей в різних положеннях, ракурсах, освітленні.
- автомобілі - шукаєш фото різних марок, кольорів, кутів.
- дефекти продукції - фото з бракованими та хорошими зразками.

Цю інформацію ми отримуємо з інтернету (Google Images, Flickr, Kaggle, Open Images Dataset, COCO, тощо). Власна зйомка (камери, смартфони). Партнери, клієнти (якщо це комерційний проєкт).

Маркування о'єктів на зображенні чи анотування Це ключовий етап: ми повинні вказати, де на зображенні знаходиться об'єкт і який у нього клас.

Розмітка включає:

Bounding box (прямокутник навколо об'єкта)

Назва класу (наприклад, person, car, dog)

Можливі й складніші анотації:

- сегментація (контур).
- ключові точки (наприклад, скелет людини).
- маски.

Інструменти для розмітки:

- LabelImg - простий, безкоштовний, локальний.
- CVAT - потужний, для великих наборів.
- Label Studio – універсальний.
- Roboflow - хмарний сервіс (підготовка, аугментація).

Формати збереження: Pascal VOC (XML), COCO (JSON), YOLO (TXT)

Підготовка датасету:

- Перевірка якості розмітки (відсутність помилок, перекриттів).
- Розподіл на навчальну, валідаційну та тестову вибірки (зазвичай: 70% / 15% / 15%).
- Форматування в той формат, який потрібен для обраної моделі.
- Аугментація (розширення даних).

Щоб модель краще вчилася, часто штучно збільшують кількість даних, для цього наприклад використовують:

- обертання.
- зміна яскравості, контрасту.
- додавання шуму.
- дзеркальне відображення.
- масштабування.

В більшості бібліотек (Ultralytics YOLO, Albumentations, Tensorflow) є готові функції для цього.

- Перевірка балансу класів.

Бажано, щоб не було сильної переваги одного класу. А кожен клас був представлений різними прикладами.

Можливі інші нюанси які полягають наприклад в:

- Висока якість зображень це бажана характеристика.
- Розміри зображень: краще, коли вони однорідні.

1.4 Другий етап попереднє опрацювання даних

Зображення, які використовуються для аналізу або розпізнавання об'єктів, часто містять різноманітні шуми небажані спотворення, які можуть знижувати точність роботи комп'ютерних алгоритмів. Джерела шуму можуть бути різними, наприклад:

- погане освітлення,
- низька якість камери,
- електронні перешкоди під час зйомки,
- стиснення зображень.

Щоб зменшити негативний вплив шуму, можна застосовувати спеціальні фільтри згладжування. Вони дозволяють зробити зображення чистішим, видалити дрібні дефекти, але при цьому намагаються зберегти важливі деталі, наприклад контури об'єктів.

До найбільш поширених методів усунення шуму належать:

Гаусове згладжування (Gaussian Blur): Метод, що розмиває зображення за допомогою математичного гаусового розподілу. Добре працює для пом'якшення дрібних випадкових шумів без суттєвої втрати контурів.

Медіанне згладжування (Median Blur): Особливо ефективно при "соляно-перцевому" шумі (чорні й білі точки). Фільтр замінює кожен піксель на медіанне значення сусідніх пікселів.

Алгоритми неймовірних середніх (Non-Local Means Denoising): Більш складні методи, які шукають схожі області на зображенні і використовують їх для усунення шуму з високою якістю.

Усунення шуму допомагає підвищити якість подальшої обробки зображень та точність розпізнавання об'єктів.

Перетворення у grayscale (відтінки сірого).

Зображення, отримані з камер чи інших пристроїв, зазвичай мають три кольорових канали: червоний, зелений та синій (RGB). Проте для багатьох задач комп'ютерного зору чи розпізнавання об'єктів вся колірна інформація не є обов'язковою.

Перетворення зображення у grayscale (відтінки сірого) полягає у зведенні трьох кольорових каналів до одного - яскравісного. У такому форматі кожен піксель характеризується лише одним числом, що показує його яскравість (від 0 це чорний до 255 це білий).

Основні переваги роботи з grayscale:

- Зменшення обсягу даних (один канал замість трьох).
- Спрощення обчислень для алгоритмів.
- Виділення структури та контурів без впливу кольору. Часто достатньо для виявлення форм, контурів та об'єктів.

Перетворення у grayscale -стандартна підготовча операція перед застосуванням методів розпізнавання контурів, сегментації, аналізу текстур та деяких моделей машинного навчання.

1.5 Третій етап виділення ознак (feature extraction)

Процес виділення ознак - це процес перетворення вхідного зображення у набір числових характеристик (ознак), які найбільш інформативно описують його вміст для задачі розпізнавання об'єктів.

Комп'ютери самі по собі не "бачать" зображення так, як люди. Для них це лише набір чисел (матриця пікселів). Завдання виділення ознак - представити зображення у вигляді характеристик, які можна використати для аналізу, класифікації, кластеризації чи детекції.

Розглянемо основну ідею:

- Знайти найважливіші риси, які добре описують об'єкт на зображенні.
- Спрощення даних - замість всієї матриці пікселів працюємо лише з ключовими числовими показниками.

Типи ознак які можуть бути використані:

- Класичні ознаки (традиційні методи комп'ютерного зору).

До появи глибоких нейронних мереж виділення ознак виконувалось вручну за допомогою спеціальних алгоритмів:

Кути та контури (Edges, Corners): Наприклад, детектори Sobel, Canny, Harris. Виділяють межі об'єктів.

Ознаки текстур (Texture Features): Наприклад, Local Binary Patterns (LBP), Гістограми Градієнтів Орієнтації (HOG).

Ключові точки (Keypoints): Наприклад, SIFT, SURF, ORB - знаходять характерні точки на об'єкті (кути, перехрестя, цікаві області), які можна порівнювати між зображеннями.

Форми та контури (Shape Features): Площа, периметр, співвідношення сторін, моменти зображення.

Ці методи особливо актуальні для задач, де кількість даних обмежена, або моделі повинні бути простими й пояснюваними.

- Автоматичне виділення ознак (глибоке навчання).

У сучасних системах, які базуються на нейронних мережах (CNN - Convolutional Neural Networks), виділення ознак відбувається автоматично:

Мережа сама навчається знаходити важливі ознаки у зображенні.

На ранніх шарах мережа виділяє прості риси: краї, текстури.

На глибших шарах - складніші: форми, частини об'єктів, орієнтації, кольорові закономірності.

Завдяки цьому моделі здатні вловлювати складні просторові та контекстні взаємозв'язки.

Це основна причина чому сучасні системи, як-от YOLO, Faster R-CNN, Detectron2 працюють так ефективно.

SVM (Support Vector Machine) - Метод опорних векторів.

Суть цього методу полягає в тому що це алгоритм класифікації (або регресії), який шукає гіперплощину (лінію в 2D, площину в 3D тощо), яка якнайкраще розділяє дані між класами.

Ключова ідея тут наступна. Побудувати межу (розділяючу площину), яка максимально віддалена від найближчих точок обох класів (опорних векторів).

Така межа буде найстійкішою до помилок на нових даних.

Використання цього підходу дуже добре працює для задач з малою кількістю ознак. Добре підходить, коли класи добре відокремлювані.

Переваги полягають у стійкості до перенавчання (overfitting).

Ефективний у високовимірних просторах (наприклад, при роботі з ознаками зображень).

Але цей підхід також має недоліки. Може бути повільним на великих наборах даних. Потребує нормалізації даних.

Його застосовують при розпізнаванні облич, класифікація тексту і при формуванні медичних діагнозів.

- KNN (k-Nearest Neighbors) це метод k найближчих сусідів.

Суть цього підходу полягає у простому алгоритмі "базованої на прикладах" класифікації.

Щоб класифікувати новий об'єкт, алгоритм шукає k- найближчих до нього прикладів з навчальної вибірки та визначає клас за більшістю серед них.

Ключова ідея полягає у схожості об'єктів які належать до одного класу.

Використовують цей підход для задач, де важлива локальна схожість.

Часто використовується для порівняння ознак, отриманих зі зображень (наприклад, після SIFT, HOG).

Він має переваги які полягають у простоті реалізації, не потребує навчання (навчальна вибірка = база знань).

Але також і має недоліки. Повільний при великій кількості даних (бо кожен раз шукає сусідів). Чутливий до вибору метрики відстані. Потребує нормалізації даних.

Його застосовують у рекомендаційних системах. При простому розпізнаванні образів. У системах виявлення аномалій.

- Decision Trees (Дерева рішень)

Суть цього підходу полягає у алгоритмі який будує дерево, де кожна внутрішня вершина - це перевірка певної ознаки (питання: наприклад, чи ознака більше або менше певного значення).

Кожна гілка веде до нових перевірок, а лист - це прогнозований клас.

Ключова ідея полягає у розбитті простору ознак так, щоб якомога краще відокремити класи.

Використання даного підходу - легко інтерпретується людиною. Може працювати з як числовими, так і з категоріальними ознаками. Переваги полягають у зрозумілому результаті (можна показати дерево). Можна виявляти важливість ознак. Не потребує нормалізації даних.

Недоліки також є і полягають у можливості легко перенавчатись (overfitting). Чутливий до зміни даних.

Застосування даного підходу може бути в медицині для формування діагнозу. При опрацюванні даних для фінансового аналізу.

Класифікація зображень (часто в комплексі з Random Forest або Gradient Boosting).

Розглянемо нижче таблицю порівнянь різних алгоритмів які можна використати для задач розпізнавання об'єктів на зображеннях.

Таблиця 1.1 – Порівняння алгоритмів для задач розпізнавання об'єктів

Алгоритм	Швидкість	Пояснюваність	Робота з великими даними	Чутливість до параметрів
SVM	повільна	середня	складно	висока
KNN	повільна	висока	складно	висока
Decision Tree	швидка	висока	добре	середня

Якщо робити підсумовуючий висновок то:

- SVM - шукає найкращу межу між класами.
- KNN - дивиться, до яких сусідів ближче.
- Decision Tree - приймає рішення через послідовні питання.

1.7 Четвертий етап розпізнавання на нових даних або Inference

Inference (інференція) - це процес, під час якого вже натренована модель застосовується до нових (раніше невідомих) зображень для передбачення або розпізнавання об'єктів.

Простіше кажучи це навчання коли модель вчиться розпізнавати. Inference коли модель вже використовує свої знання, щоб робити передбачення.

Працює inference при розпізнаванні об'єктів наступним чином:

- Отримання нового зображення.
1. Попередня обробка зображення (як ми вже розглядали):
 - Зменшення шуму,
 - Зміна розміру (під розмір, який очікує модель),
 - Нормалізація пікселів,
 2. Перетворення у потрібний формат (наприклад, тензор).
 3. Передача зображення у модель.
 4. Модель проводить обчислення й видає результат:

- у випадку класифікації — ймовірності кожного класу,
- у випадку детекції — координати об'єктів, назви класів, впевненість.

5. Постобробка:

- фільтрація за порогом впевненості,
- видалення зайвих об'єктів (наприклад, Non-Max Suppression для детекції).

6. Візуалізація результату (нанесення прямокутників, підписів тощо).

1.7 Етап оцінки якості розпізнавання. Використання метрик точності

Метрики для класифікації:

1. Ассурасу (Точність класифікації). Ассурасу (Точність класифікації) - це один з основних показників ефективності класифікаційної моделі. Він показує, яка частина усіх передбачень виявилася правильними.

$$\text{Accuracy} = \frac{TP+TN}{FP+FN+TP+TN} \quad (1.1)$$

Де:

TP (True Positive) - істинно позитивні (правильне передбачення класу "позитив")

TN (True Negative) - істинно негативні (правильне передбачення класу "негатив")

FP (False Positive) - хибно позитивні (неправильне передбачення "позитив" замість "негатив")

FN (False Negative) — хибно негативні (неправильне передбачення "негатив" замість "позитив")

Ассурасу має недолік - погано працює при незбалансованих даних (коли одного класу значно більше).

2. Precision (Точність для конкретного класу)

Precision показує, яку частину всіх передбачень для певного класу модель зробила правильно.

$$\text{Precision} = \frac{TP}{TP+FP} \quad (1.2)$$

Де:

TP (True Positive) — істинно позитивні (модель правильно визначила клас)

FP (False Positive) — хибно позитивні (модель помилково віднесла до класу)

Якщо модель дозволяє виявляти рак на медичних зображеннях, важливо, щоб вона рідко помилялась, і не видавала здорових людей за хворих.

3. Recall (Повнота, чутливість)

$$\text{Recall} = \frac{TP}{TP+FN} \quad (1.3)$$

Де:

TP (True Positive) - істинно позитивні (правильно передбачені)

FN (False Negative) - хибно негативні (пропущені)

Важливо не пропустити хворих пацієнтів.

4. F1-score (Збалансована метрика)

$$F1 = 2 \cdot \frac{\text{Precision} \cdot \text{Recall}}{\text{Precision} + \text{Recall}} \quad (1.4)$$

Компроміс між Precision та Recall. Використовується, коли важливий баланс між пропущеними та хибно позитивними.

Метрики для локалізації об'єктів (Bounding Boxes)

5. IoU (Intersection over Union)

IoU - основна метрика для оцінки якості локалізації об'єктів (bounding boxes - прямокутників) у задачах детекції.

$$\text{IoU} = \frac{\text{Площа об'єднання}}{\text{Площа перетину}} \quad (1.5)$$

Визначає, наскільки точно передбачена рамка збігається з реальною.

Метрики для комплексної оцінки моделей детекції.

7. mAP (mean Average Precision)

mAP (середня точність) - це основна комплексна метрика якості роботи моделей об'єктної детекції (YOLO, Faster R-CNN, SSD тощо).

Це середня точність для всіх класів, з урахуванням різних порогів IoU.

mAP широко використовується для оцінки моделей обробки зображень (YOLO, Faster R-CNN тощо).

Як розраховується mAP:

Розраховують Precision-Recall криву для кожного класу.

Визначають середню точність (AP - Average Precision) для кожного класу.

Усереднюють по всіх класах та отримують mAP.

Таблиця 1.2 - Порівняння метрик

Метрика	Де використовується	Навіщо
Accuracy	Класифікація	Загальна точність
Precision	Класифікація	Чи часто модель помиляється в "позитивах"
Recall	Класифікація	Чи пропускає модель позитиви
F1-score	Класифікація	Баланс між Precision та Recall
IoU	Детекція	Наскільки добре рамка накрила об'єкт
mAP	Детекція	Загальна якість виявлення об'єктів

Під етап аналізу помилок та донавчання моделі (Fine-tuning).

1. Аналіз помилок (Error Analysis)

Аналіз помилок дозволяє зрозуміти де саме модель помиляється, чому модель робить ці помилки, як можна покращити якість роботи.

Розглянемо основні кроки аналізу помилок:

1. Побудова матриці помилок (Confusion Matrix).

Показує, скільки разів кожен клас був передбачений правильно чи неправильно. Добре підходить для класифікації.

2. Перегляд конкретних прикладів.

Знайти зображення, на яких модель дала неправильний результат.

Візуально проаналізувати – можливо тут погана якість зображень, некоректна розмітка, дуже схожі класи, складні умови (освітлення, пози, часткове перекриття).

3. Визначення типів помилок:

- False Positives (помилково передбачений об'єкт).
- False Negatives (модель не виявила об'єкт).
- Низька точність локалізації (низький IoU).

4. Аналіз складних випадків (edge cases):

- Дуже малі об'єкти.
- Різноманітні масштаби.
- Нечіткість об'єктів.

2. Донавчання моделі (Fine-tuning).

Коли ви знаєте, де помиляється модель - можна її донавчити.

Що ж таке fine-tuning? Це додаткове навчання вже існуючої моделі на нових або виправлених даних.

Використовуються додаткові або складніші приклади.

Може бути повне донавчання або часткове (оновлюємо лише частину шарів нейромережі). Fine-tuning потрібен для покращення якості роботи на складних даних. Адаптувати під нові типи об'єктів. Виправити помилки на конкретних підгрупах.

Для цього необхідно зробити:

1. Збір додаткових даних:

- Зібрати більше прикладів там, де були помилки.
- Зробити додаткову розмітку.

2. Аугментація даних (Data Augmentation):

- Розширити тренувальний набір за допомогою:
- поворотів,
- масштабування,
- зміни яскравості,
- шуму,
- зсувів тощо.

3. Зменшити швидкість навчання (learning rate):

- Fine-tuning зазвичай робиться на малих значеннях learning rate.

4. Використати pre-trained модель:

Наприклад, взяти попередньо натреновану модель YOLO, ResNet, EfficientNet і донавчити її.

Серед розглянутих технологій та етапів опрацювання розглянемо більш детальніше типові методи які використовуються для розпізнавання.

Головна перевага класичного CV полягає у використанні для простих задач без потреби у великих даних.

Головний недолік полягає у гіршому опрацюванні складних сцен та великою варіативністю зображень.

CNN - це глибокі нейронні мережі, які самі вчаться виділяти ознаки із зображення, без необхідності вручну задавати, що саме шукати (на відміну від класичного CV).

Замість ручного проектування ознак, CNN отримує "сирі" пікселі зображення і, проходячи через багато шарів, автоматично навчається розпізнавати:

- на початкових шарах - прості патерни: контури, кути, текстури;
- на середніх - фрагменти об'єктів: очі, вуха, колеса тощо;
- на глибоких - цілісні образи: людина, кіт, машина тощо.

Кожен наступний шар бачить все більш абстрактні представлення об'єкта.

VGG - одна з ранніх популярних CNN-архітектур, названа на честь Visual Geometry Group (Оксфорд).

Її головна ідея простота та однорідність:

- усі згорткові шари (convolution) мають маленький розмір ядра 3×3 ;
- після кількох згорток стоїть операція зменшення розміру (max pooling);
- після великої кількості згорткових шарів йдуть повнозв'язні шари (fully connected);
- глибина мережі - наприклад, у VGG-16, є 16 шарів.

Суть VGG - не у складності, а в глибокій послідовності простих операцій.

Переваги VGG легко розуміти і реалізовувати, добре працює для задач класифікації. Окрім переваг є недоліки VGG велика кількість параметрів (мільйони), потребує великої обчислювальної потужності і багато пам'яті.

ResNet (Residual Network) - архітектура, яка зробила революцію в глибокому навчанні.

Основна проблема дуже глибоких мереж - зі збільшенням кількості шарів часто погіршується точність (через зникнення градієнтів під час навчання). ResNet вирішує цю проблему завдяки резидентним (залишковим) зв'язкам.

Проста ідея ResNet додається короткий прямий зв'язок ("skip connection") між шарами: мережа вчиться не весь шар, а лише "різницю" між вхідним і правильним виходом. Це дозволяє будувати дуже глибокі мережі (50, 101, 152 шари і більше) без проблем з навчанням.

Переваги ResNet полягають у тому що вона дозволяє будувати дуже глибокі моделі без втрати якості, краще узагальнює дані, стала базою для багатьох сучасних моделей. Недоліки ResNet пов'зані з трохи складнішою структурою для розуміння, великі мережі все одно потребують багато обчислювальних ресурсів.

Підсумовуючи можна зробити висновок:

VGG проста і глибока послідовність шарів без додаткових зв'язків.

ResNet - глибша мережа, яка використовує залишкові зв'язки для полегшення навчання.

YOLOv8 - це одна з найновіших та найпотужніших версій лінійки YOLO (You Only Look Once).

Основна ідея всієї лінійки YOLO полягає у тому що модель бачить зображення лише один раз (one pass) і одразу видає координати об'єктів і класи.

Особливості YOLOv8 це вже чисто нейронна архітектура без "ручного" підбору ознак. Вона повністю end-to-end: подаємо зображення і отримуємо бокси і класи.

Має гнучку архітектуру для класифікації, сегментації, трекінгу та об'єктної детекції.

Структура ще більш оптимізована, ніж у попередніх YOLOv5/YOLOv7:

- краще працює з маленькими об'єктами;
- краще узагальнює дані;
- підтримує різні розміри моделей (small, medium, large).

Дуже швидко працює в реальному часі, навіть на слабких GPU чи мобільних пристроях.

Використовує покращену головну частину моделі CSPDarknet а також нові шляхи оптимізації.

Переваги YOLOv8:

- Дуже висока точність і швидкість.
- Можна запускати на edge-девайсах.
- Підтримка сучасних фреймворків (PyTorch).

Підтримує transfer learning - можна швидко донавчити на своїх даних.

Але також має недоліки YOLOv8:

- Потребує достатнього об'єму даних для якісного навчання.
- Висока чутливість до якості анотацій.

SSD - це також метод об'єктної детекції, який конкурував свого часу з YOLO, особливо у версіях SSD300, SSD512.

Основна ідея SSD це замість того, щоб розбивати зображення на сітку, як у класичному YOLO, SSD напряду прогнозує бокси на кількох рівнях масштабів.

Використовує декілька шарів для знаходження як дрібних, так і великих об'єктів.

На кожному шарі генерується певна кількість "якорних" (anchor) боксів різних розмірів і пропорцій.

Це дає хорошу продуктивність для об'єктів різних масштабів.

Переваги SSD полягають у хорошій баланс між точністю та швидкістю, легше масштабується під великі чи маленькі об'єкти.

Може працювати в реальному часі на більшості GPU.

Легше тренувати, ніж двоетапні моделі (типу Faster R-CNN).

Недоліки SSD є гірша точність на маленьких об'єктах, ніж у сучасних YOLO чи Faster R-CNN.

Зараз поступово витісняється новішими YOLOv5, v7, v8, DETR.

SD - класичніша модель (з 2016 року), підходить для багатьох задач, але поступається новим методам за точністю.

YOLOv8 - сучасний універсальний флагман для об'єктної детекції: ще точніший, ще швидший, більш оптимізований для сучасних задач.

SSD - "перше покоління" швидких одноетапних детекторів. YOLOv8 - "сьоме-восьме покоління" оптимізованих моделей.

Mask R-CNN - це розширення популярного алгоритму Faster R-CNN, який крім детекції об'єктів (бокси та класи), додає ще й піксельну сегментацію кожного об'єкта.

Тобто модель не просто малює рамку навколо об'єкта, а виділяє точний контур (маску) об'єкта на зображенні.

Основні переваги Mask R-CNN це дуже точна сегментація піксель за пікселем. Може працювати з перекриттям об'єктів. Універсальна: застосовується у медицині, автономному водінні, контролі якості.

Недоліки полягають у тому що вона потребує більше обчислювальних ресурсів, ніж прості детектори (SSD, YOLO). Навчання потребує детальної розмітки (маски кожного об'єкта). Швидкість нижча, ніж у YOLO для реального часу.

Застосовується переважно в медичні знімки, автономне водіння, контроль виробництва, відеоаналітика, біометричні системи.

Основна ідея полягає у застосуванні SAM - це зовсім нове покоління сегментації (розробка Meta, 2023 рік).

На відміну від Mask R-CNN, який вчиться на конкретних класах, SAM створено для універсальної сегментації будь-чого:

- працює без донавчання на нових об'єктах;
- вміє сегментувати практично будь-які об'єкти на будь-якому зображенні, навіть ті, яких не бачив під час навчання.

Працює SAM наступним чином приймає вхід у вигляді зображення та підказок (prompt): точок, боксу, контурів або текстового опису.

Після цього видає точну маску об'єкта.

Може працювати і без підказок - у режимі автоматичної сегментації.

Переваги SAM - не прив'язаний до конкретних класів об'єктів. Надзвичайно точна сегментація. Працює навіть з дуже складними сценами.

Дуже універсальний справжня система загальної сегментації.

Недоліки - дуже великі моделі (багато параметрів). Потребує потужного GPU для роботи в повноцінному режимі. Не завжди стабільно працює на складних семантичних задачах (плутанина з фоном чи схожими об'єктами).

Застосовується для AI-редактори зображень, генеративні моделі, агротех (виділення рослин), медицина. розмітка великих датасетів.

Загальний висновок може бути наступним.

Mask R-CNN - класична потужна модель для навченої сегментації по класах (потребує підготовлених масок на етапі навчання).

SAM - універсальна zero-shot сегментація без потреби навчати під конкретні об'єкти (працює з будь-якими об'єктами практично "з коробки").

Основна ідея Vision Transformers (ViT)

Vision Transformers - це новий підхід до обробки зображень, який прийшов з області обробки текстів (NLP), де вже давно домінували трансформери (наприклад, BERT, GPT). Замість того, щоб використовувати згорткові мережі (CNN), ViT працює з зображенням так само, як GPT працює з текстом.

Працює Vision Transformer наступним чином зображення ділиться не на пікселі чи фільтри, а на маленькі шматочки - патчі (наприклад, 16x16 пікселів).

Кожен патч перетворюється в вектор (ембедінг) - тобто отримує представлення в числовому вигляді.

Всі ці патчі формують послідовність - аналогічно до послідовності слів у реченні.

Далі вся послідовність патчів подається у трансформер - блоки self-attention, які дозволяють кожному патчу бачити всі інші патчі.

На виході - загальне представлення зображення, яке можна подавати у класифікатор або для іншої задачі.

Головна відмінність від CNN полягає у тому що у CNN є локальні фільтри: кожен шар "бачить" спочатку лише маленьку ділянку зображення.

У ViT кожен патч може одразу взаємодіяти з усіма іншими патчами через self-attention - це дозволяє моделі враховувати глобальний контекст з самого початку.

Переваги Vision Transformers дуже добре працюють на великих обсягах даних. легко масштабуються до надвеликих моделей, краще захоплюють глобальні зв'язки в зображенні.

Можуть поєднуватися з текстом, мовою, аудіо - універсальні архітектури (основа багатьох мультимодальних моделей, наприклад, CLIP, Flamingo, DINOv2, SAM тощо).

Дозволяють отримувати більш інтерпретовані attention карти.

Недоліки Vision Transformers - потребують великих обсягів даних для гарного навчання (тисячі або мільйони зображень). Важчі для навчання "з нуля"

без попереднього навчання. На малих датасетах часто поступаються CNN. Потребують більше пам’яті та обчислювальних ресурсів.

У таблиці 1.3 зведені описані порівнянні методи які актуальні на 2025 рік.

Таблиця 1.3 - Порівняння типових методів для розпізнавання об’єктів на зображенні

Метод	Актуальність станом на 2025	Де застосовується
Класичний CV	Обмежена	Просте відео, робототехніка
CNN (ResNet, VGG)	Середня	Базова класифікація
YOLOv8, SSD	Висока	Об’єктне детектування
Mask-RCNN, SAM	Дуже висока	Сегментація, медицина
Vision Transformers	Найсучасніше	Мультимодальні системи, AI

Таким чином провівши порівняльний аналіз цікавою моделлю для задач розпізнавання об’єктів на зображенні є YOLO v8.

1.8 Висновок до першого розділу

У першому розділі кваліфікаційної роботи бакалавра описано етапи які використовуються під час розпізнавання об’єктів на зображенні. Наведено порівняльний аніліз алгоритмів для задач розпізнавання, наведені метрики які вкажуть на якість розпізнавання, здійсненл їх порівняльний аналіз. Окрім цього здійснено порівняльний аналіз моделей нейронних мереж для задач розпізнавання об’єктів на зображенні.

РОЗДІЛ 2. РОЗРОБКА ПРОГРАМИ ДЛЯ РОЗПІЗНАВАННЯ ОБ'ЄКТІВ НА ЗОБРАЖЕННІ

2.1 Переваги та аналіз версії нейронної мережі YOLO v8

Розглянемо що таке YOLOv8. YOLOv8 - це найновіша версія в родині YOLO (You Only Look Once), яку випустила компанія Ultralytics ще у 2023 році.

Вона є суттєвим кроком вперед порівняно з попередніми версіями YOLOv5 чи YOLOv7:

- точніше працює;
- гнучкіша у використанні;
- підтримує більше задач.

Основні новації YOLOv8:

- Уніфікована архітектура
- YOLOv8 — це універсальна модель може виконувати не тільки об'єктну детекцію, а й класифікацію, сегментацію (instance segmentation), трекінг, позначення ключових точок (keypoint detection).

Нова архітектура Backbone використовується вдосконалена версія CSPNet (CSPDarknet не використовується напряду як у YOLOv5). Більш легка та ефективна структура моделі. Вбудовані механізми глибокої оптимізації розмірів параметрів.

Зменшення кількості параметрів завдяки оптимізаціям модель має менше параметрів порівняно з аналогічними YOLOv5/7 при збереженні або навіть підвищенні точності.

Гнучка підтримка розмірів моделей є кілька версій:

- YOLOv8n (nano);
- YOLOv8s (small);
- YOLOv8m (medium);
- YOLOv8l (large);
- YOLOv8x (extra large).

Можна вибрати баланс між швидкістю та точністю. Сучасний тренувальний пайплайн підтримує:

- data augmentation (на льоту),
- Label smoothing,
- Mosaic, MixUp,
- EMA (Exponential Moving Average),
- автоматичний early stopping,
- інтеграцію з TensorRT, ONNX, OpenVINO.

Розглянемо основні технічні переваги YOLOv8.

Таблиця 2.1 – переваги застосування YOLOv8

Перевага	Пояснення
Висока точність	Краще знаходить дрібні об'єкти
Висока швидкість	Працює в реальному часі
Легка адаптація	Просте донавчання на своїх даних
Портативність	Можна запускати на CPU, GPU, TPU, мобільних пристроях
Легкість візуалізації	Зручні інструменти в Ultralytics API
Постійна підтримка	Активно розвивається спільнотою Ultralytics

Якщо розглянути більш детальніше версії YOLOv8.

YOLOv8n (nano). Основна задача це швидкість і легкість.

Підходить для мобільних пристроїв, embedded систем камер відеонагляду з обмеженими ресурсами, простих задач з не дуже складними сценами.

Має дуже мало параметрів можна запускати навіть без потужного GPU.

Наприклад: ~3 млн параметрів, висока швидкість inference. Недоліком є найменша точність із усієї лінійки.

YOLOv8m (medium). Компроміс між точністю та швидкістю. Підходить для систем відеоспостереження, промислових задач, проектів зі стандартними GPU, більшості задач розпізнавання, де потрібна хороша якість.

Має більшу здатність виявляти дрібні об'єкти порівняно з nano.

Приблизно ~25 млн параметрів, середній баланс споживання пам'яті.

Суттєво вища точність, ніж у nano.

Таблиця 2.2 – Типи версій YOLOv8

Назва	nano (n)	medium (m)	extra-large (x)
Мета	Дуже швидкий, для слабких пристроїв	Компроміс між точністю і швидкістю	Максимальна точність
Кількість параметрів	Найменша	Середня	Найбільша
Потреба в пам'яті	Мінімальна	Помірна	Висока
Точність	Найнижча	Середня	Найвища
Швидкість обробки	Максимальна	Помірна	Нижча

YOLOv8x (extra-large). Максимальна точність, максимум можливостей.

Підходить для досліджень, медицини, задач зі складною сценою (багато дрібних і перекритих об'єктів), високоточних систем контролю якості.

Потребує потужного GPU з великою пам'яттю.

Наприклад, ~68 млн параметрів, вимагає багато відеопам'яті.

Найбільш точне виявлення.

Таблиця 2.3 – Вимоги до версій YOLOv8

Версія	Params (млн)	Відеопам'ять (640x640)	FPS (на RTX 3090)
YOLOv8n	~3	~1.2GB	500+
YOLOv8m	~25	~3.5GB	~150
YOLOv8x	~68	~8GB	~50-60

Чим більша літера тим потужніша, точніша, важча і повільніша модель.

2.2 Попередня обробка

Для нашого програмного забезпечення будемо працювати наприклад з зображенням поданими на рисунку 2.1. Для виявлення на зображенні об'єктів потрібно здійснити попередню обробку.



а)

б)

Рисунок 2.1. – Оригінальне зображення

Проведемо етапи попереднього опрацювання для цього застосуємо ресайзінг зображення.

1. Ресайз зображення.

Ресайз зображення - це зміна розміру зображення по ширині та висоті, тобто ми переводимо зображення з його оригінального розміру в новий, заданий розмір.

Наприклад, було зображення 1920×1080 пікселів. Після ресайзу воно стає 640×640 пікселів.

Навіщо порібно роблять ресайз?

Уніфікація вхідних даних.

Більшість нейронних мереж очікують фіксований розмір вхідного зображення.

Наприклад, YOLOv8 зазвичай працює з квадратами: 640×640 , 512×512 тощо. Якщо подавати різні розміри модель не зможе коректно обробити.

Тому нам буде потрібно робити ресайз досліджуваних зображень.

2. Прискорення обробки.

Маленькі зображення обробляються швидше.

Відеоаналітика, розпізнавання в реальному часі дуже залежні від швидкості.

3. Оптимізація використання пам'яті

Великі зображення споживають багато GPU-пам'яті. Зменшення розміру зменшує потребу в ресурсах.

4. Уникнення перевчання на деталях.

Іноді високі роздільності містять шум або надлишкові деталі. Ресайз дозволяє моделі фокусуватись на суттєвих об'єктах.

Розглянемо види ресайзу.

Простий ресайз (звичайна зміна розміру) приклад коду подано на рисунку 2.2.

```
import cv2
resized_img = cv2.resize(img, (640, 640))
```

Рисунок 2.2 – Фрагмент коду звичайна зміна розміру

Просто масштабує зображення в новий розмір.

Недолік - може спотворити пропорції об'єктів, якщо зображення не квадратне.

Letterbox (збереження пропорцій).

Масштабує зображення до максимально можливого розміру в межах нової рамки, а порожні області заповнює (падінгом).

Використовується в YOLOv8 за замовчуванням: Приклад застосування подано на рисунку 2.3.

```
from ultralytics.utils import ops
img_resized = ops.letterbox(img, new_shape=(640,
640))
```

Рисунок 2.3 – Фрагмент коду звичайна зміна розміру

Таким чином пропорції зберігаються, не викривляються об'єкти, додаткові пусті області просто заповнюються (наприклад, сірим кольором).

Наслідок неправильного ресайзу можуть бути якщо сильно деформувати співвідношення сторін - модель може:

- Не впізнати об'єкти;
- Неправильно визначити їхні контури;
- Знизити точність роботи.

Після того як здійснена попередня обробка зображень нам необхідно провести наступний крок тобто коли ми запускаємо вже навчену модель на нових зображеннях.

Зображення після підготовки.

Для inference (тобто коли ми запускаємо вже навчену модель на нових зображеннях) зазвичай робляться тільки ресайз, нормалізація.

Нормалізація зображення - це процес приведення значень пікселів до певного стандартного діапазону перед подачею у нейромережу.

1. Щоб зробити значення пікселів зручними для обробки моделлю.
2. Щоб прискорити і стабілізувати навчання (оскільки багато алгоритмів оптимізації працюють краще, коли вхідні дані мають однаковий масштаб).
3. Щоб уникнути проблем з числовою нестабільністю (переповненням, дуже великими градієнтами).

Як виглядають значення пікселів до нормалізації?

Стандартне зображення - це матриця значень пікселів від 0 до 255 (тобто 8-бітна глибина кольору).

Перетворення у правильний колірний простір передбачає кодування у певній системі представлення кольорів. Найпоширеніші: YCrCb, LAB, CMYK, GRAY, HSV, RGB, BGR

Чому виникає проблема? Тому що більшість моделей (YOLOv8, ResNet, VGG) навчаються на зображеннях у форматі RGB.

Але OpenCV за замовчуванням читає зображення у форматі BGR.

Це важливо тому що якщо подати модель неправильною послідовністю каналів - вона буде плутати кольори.

Наприклад, замість червоного побачить синій. Замість білого - фіолетовий.

Результати детекції сильно погіршуються.

Як виправити - перетворення BGR в RGB. Для цього можна використати приклад на рисунку 2.4

```
cv2.cvtColor()
```

Рисунок 2.4 – Фрагмент коду перетворення BGR в RGB

Для YOLOv8 це робиться автоматично. Якщо використовувати стандартний API Ultralytics YOLOv8 - він сам переводить у RGB всередині себе. Якщо готуються дані вручну то треба пам'ятати про це і переводити самому.

Тензоризація.

Тензоризація - це перетворення зображення з формату, який розуміє людина (наприклад, масив NumPy), у формат, який "розуміє" нейронна мережа тобто у тензор (багатовимірний масив).

Усі сучасні фреймворки (PyTorch, TensorFlow) працюють з даними у вигляді тензорів.

Тензори - це розширення поняття матриць:

0D: скаляр

1D: вектор

2D: матриця (зображення чорно-біле)

3D: кольорове зображення (висота $\times$ ширина $\times$ канали)

4D: батч зображень (кількість зображень $\times$ канали $\times$ висота $\times$ ширина)

Типова форма тензора зображення. Зображення 640 $\times$ 640, 3 канали (RGB)

Типови сценарій робить і нензоризацію при використанні YOLOv8.

Наприклад код нижче поданий на рисунку 2.5

```
from ultralytics import YOLO
model = YOLO('yolov8n.pt')
results = model.predict('image.jpg')
```

Рисунок 2.5 – Фрагмент коду для тензоризації

Достягнути передати просто шлях до файлу.

Далі Ultralytics, читає картинку, переводить у RGB, робить ресайз, нормалізацію, переводить у тензор, додає батч-розмірність, подає в модель. На рисунку 2.6 наведені результати попередньої обробки які необхідні для YOLOv8.



а)

б)

Рисунок 2.6 – Зображення після підготовки

Після застосування попередньої обробки слідує наступний крок застосування карток ознак.

2.3 Застосування карти ознак (feature maps)

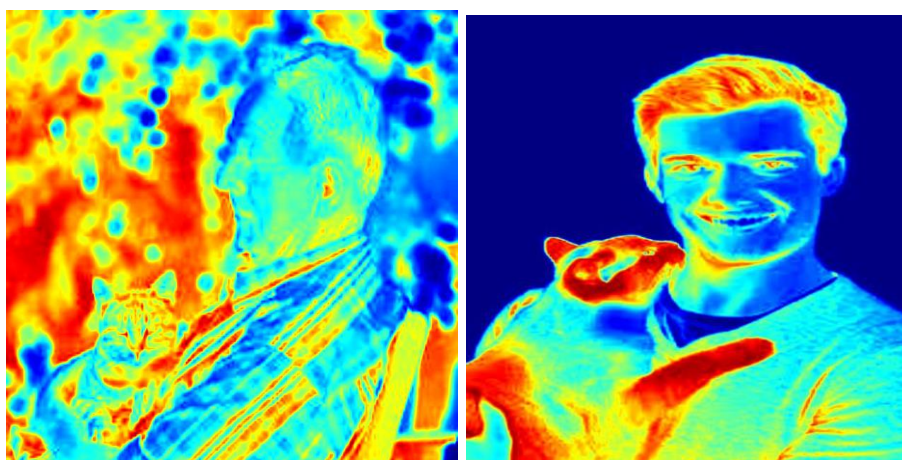
Що таке feature map (картки ознак). Коли зображення проходить через неймережу, кожен її шар виділяє певні ознаки. Карта ознак - це результат роботи фільтрів на кожному конкретному шарі.

Кожен канал карти ознак показує, на які патерни реагує мережа:

- контури;
- текстури;
- орієнтації;
- форми;
- більш складні деталі об'єктів на глибших шарах.

На рисунку 2.7 подані умовні зображення, які імітують карти ознак на різних етапах обробки зображення в CNN / YOLOv8 для двох рисунків:

Після застосування глибокі шари - витягують складніші ознаки. Чим далі - тим абстрактніші feature maps.



а)

б)

Рисунок 2.7 – Зображення карти ознак (feature maps)

Після отримання результатів на цьому етапі необхідно розпізнати об'єкти на зображенні для цього використовуємо наступний етап.

2.4 Зображення розпізнаних об'єктів (Inference)

На етапі розпізнавання об'єктів (inference) при застосуванні YOLOv8 відбувається швидке і точне виявлення об'єктів на зображенні або кадрі відео в реальному часі. Розглянемо, що саме робить модель під час inference (інференсу):

Зображення подається до моделі. Де воно масштабується до стандартного розміру (наприклад, 640×640 пікселів).

Виконується нормалізація: значення пікселів перетворюються до діапазону $[0, 1]$ або $[-1, 1]$.

Зображення конвертується в тензор та змінюється порядок каналів (як правило, з HWC у CHW).

Пропуск зображення через нейронну мережу (forward pass).

Зображення проходить через мережу YOLOv8, яка складається з:

Backbone - витягує базові ознаки (feature maps).

Neck - комбінує ознаки на різних масштабах.

Head - формує остаточні передбачення: координати, класи та ймовірності.

Отримання «сирих» передбачень.

На виході модель дає набір прямокутників (bounding boxes) з такими даними для кожного з них:

x, y, w, h - координати та розміри прямокутника.

conf - впевненість моделі, що це об'єкт.

class_probabilities - ймовірності приналежності до кожного класу (людина, кішка, машина тощо).

class_id - індекс класу з найбільшою ймовірністю.

Фільтрація результатів.

Виконується кілька перевірок:

Поріг впевненості (confidence threshold) - залишаються лише ті об'єкти, для яких $\text{conf} > 0.25$ (або інше значення).

Non-Maximum Suppression (NMS) - щоб прибрати дублікати, якщо модель виявила кілька прямокутників для одного об'єкта. Залишається лише один із найвищим score.

Постобробка та масштабування назад.

Координати прямокутників масштабуються назад до оригінального розміру зображення.

Формується фінальний список передбачень: координати, клас, назва, ймовірність.

Візуалізація результатів (опційно)

Прямокутники малюються на зображенні.

Додаються підписи: клас об'єкта та відсоток впевненості.

```
from ultralytics import YOLO

# Завантажити модель
model = YOLO('yolov8n.pt')

# Запустити розпізнавання (inference) на зображенні
results = model('image.jpg')

# Вивести результати
results.show()

results.save()          # зберегти зображення з
прямокутниками
```

Рисунок 2.8 – Фрагмент коду для тензоризації



Рисунок 2.9 – Зображення розпізнаних об'єктів (Inference)

Як ми бачимо при використанні YOLOV8n На рисунку 2.9а було добре розпізнано такі об'єкти як кіт та людина. При застосування YOLOV8n до іншого зображення результат гірший, оскільки вдалося розпізнати лише людину. При застосуванні YOLOV8m результат розпізнавання кращий (див. Рис. 2.9в).

2.5 Розпізнавання об'єктів на зображенні та виділення їх контуром

При розпізнавання об'єктів на зображенні та виділення їх контуром при застосуванні моделей YOLOv8n, YOLOv8m, YOLOv8x відбувається розпізнавання об'єктів на зображенні та виділення їх контуром .

Модель автоматично масштабує зображення, обчислює ознаки, знаходить прямокутники навколо об'єктів, повертає список знайдених об'єктів з:

- координатами прямокутника (bounding box),
- назвою класу (людина, кіт, тощо),
- впевненістю (наприклад, $0.85 = 85\%$).

Візуалізація результатів (виділення контуром).

YOLO автоматично малює прямокутник навколо кожного об'єкта дає назву назву та відсоток впевненості:

якщо потрібно отримати координати то можна використати код поданий на рисунку 2.8.

```
for box in results[0].boxes:
    print(box.xyxy, box.conf, box.cls)
```

Рисунок 2.8 – Фрагмент коду для тензоризації

Приклад результату отриманого н зображенні:

Прямокутник навколо людини або об'єкта.

Надпис person 0.40, 0.26 подано на рисунку 2.11а.



Рисунок 2.11 – Зображення та розпізнані об'єкти при застосуванні різних версій YOLOv8

YOLOv8n, m, x дозволяють виконувати розпізнавання об'єктів і автоматично обводити їх прямокутниками з підписами. Різниця між ними — у швидкості та точності. Приклади застосування розробленої програми з різними версіями YOLOv8n, m, x наведено у додатках. Із проведених експериментів можна рекомендувати використання для розпізнавання таких об'єктів на зображенні як кіт і людина саме версію YOLOv8x, оскільки саме вона показала найкращий результат.

2.6 Висновок до другого розділу

В даному розділі кваліфікаційної роботи описані різні типи моделей для розпізнавання об'єктів на зображенні YOLOv8. Описані та продемонстровано роботу на кожному з етапів під час проведення експериментів по розпізнаванню об'єктів таких як кіт а т людина на різних зображеннях. Вони показали що при застосуванні різних версій YOLOv8n, m, x найкращий результат давала версія саме YOLOv8x. Тому її можна рекомендувати саме для задач розпізнавання таких об'єктів.

РОЗДІЛ 3. БЕЗПЕКА ЖИТТЄДІЯЛЬНОСТІ, ОСНОВИ ОХОРОНИ ПРАЦІ

3.1 Безпека життєдіяльності. Мета та завдання

Вся сукупність видів людської активності утворює поняття діяльності. Якраз діяльність і вирізняє людину від інших живих істот, вона є специфічно людською формою активності, необхідною умовою існування людського суспільства. Форми діяльності розмаїті. Вони охоплюють практичні, інтелектуальні і духовні процеси, які протікають в побуті, громадській, культурній, виробничій, науковій та інших сферах життя.

Діяльністю займаються всі – діти, дорослі, люди похилого віку, тому безпека діяльності має відношення до всіх людей. Небезпеки підстерігають людей не тільки на виробництві, тому вивчення лише виробничого травматизму в системі загальної безпеки життєдіяльності не висвітлює проблеми.

Безпека – це стан діяльності, при якому з певною ймовірністю виключається прояв небезпек. Безпека – це мета, а безпека життєдіяльності – засоби, шляхи, методи її досягнення.

Актуальність дисципліни ще більше зростає у зв'язку з існуванням аксіоми про потенційну небезпеку діяльності: в жодному виді діяльності неможливо досягнути абсолютної безпеки, будь-яка діяльність потенційно небезпечна.

Завдання БЖД є розробка методів прогнозування, вивчення та ідентифікації шкідливих факторів, їх впливу на людину і довкілля.

Курс БЖД призначений:

- сприяти усвідомленню, що в центрі уваги повинна бути людина, як головна цінність суспільства, та виховати в людині гуманне, свідоме
- ставлення до питань особистої безпеки та безпеки оточуючих в усіх сферах відносин;
- виробити навички ідентифікації небезпечних та шкідливих факторів і створення сприятливих умов життєдіяльності людей на певній території;

- тримати на контролі проектування нової техніки і технологічних процесів згідно з сучасними вимогами екології і з урахуванням стійкості функціонування господарських об'єктів та технічних систем;
- прогнозувати можливу обстановку і приймати грамотні рішення в умовах надзвичайних ситуацій щодо захисту населення та персоналу об'єктів від можливих негативних наслідків;
- забезпечити якісне засвоєння нового стереотипу поведінки людини з метою виживання в нових природних та антропогенних умовах.

Безпека життєдіяльності базується на досягненнях таких дисциплін, як інженерна психологія, фізіологія людини, охорона праці, екологія, ергономіка, економіка тощо. Вона була і є в центрі уваги людей. З древніх часів до наших днів людина прагнула забезпечити свою безпеку. З розвитком промисловості це потребує спеціальних знань. БЖД особливо актуальна зараз, в добу науково-технічного прогресу. Вона покликана відіграти важливу роль в стабілізації людського суспільства.

Завдання курсу “Безпека життєдіяльності” (БЖД) полягає у чіткому розумінні небезпечних чинників у ситуаціях, що виникають як у середовищі проживання людини, так і у середовищі навчання і праці.

БЖД – це ступінь захисту людини від надзвичайної небезпеки, де під терміном “небезпека” мається на увазі вплив на людину факторів, які можуть викликати відхилення стану її здоров'я від нормального. Природа цих факторів може бути пов'язана як з причинами природного або соціально-економічного характеру (екологічними катастрофами, низьким економічним рівнем життя та ін.), так і з причинами техногенного характеру (з рівнем забруднення навколишнього середовища як наслідком виробничої діяльності людини, аваріями, катастрофами на підприємстві, транспорті, війнами та ін.).

Викладання дисципліни має на меті:

- вивчення структури, змісту і взаємозв'язку життєдіяльності людини із середовищем праці й проживання;

- визначення чинників, причин і параметрів, що сприяють виникненню надзвичайних ситуацій;
- визначення принципів і способів захисту людей в умовах повсякденного життя, а також в умовах надзвичайних ситуацій.

3.1 Інформаційне забезпечення БЖД

Відповідно до ДСТУ 2938-94 "Системи обробки інформації. Основні поняття. Терміни та визначення" комп'ютер — це функційний пристрій, що складається з одного чи кількох взаємопов'язаних центральних процесорів і периферійних пристроїв і може виконувати обчислення без участі людини

Основними функціями комп'ютера є введення та виведення інформації, її зберігання та обробка. В якості пристроїв введення часто використовуються клавіатура та сканер, який забезпечує більшу швидкодію. Інформацію також можна вводити шляхом її зчитування з магнітних, оптичних та оптико-магнітних носіїв. Комп'ютер може отримувати інформацію і з комп'ютерної мережі.

Введена в системний блок інформація впорядковується або опрацьовується відповідно до програми („програмне забезпечення”), яке визначає логічні кроки процесу опрацювання. Цей процес повністю автоматизований і здійснюється без зовнішнього впливу.

Пристроями виведення можуть бути дисплеї, друкувальні (принтер) та графопобудовуючі (плотер) пристрої. Інформація може також виводитись на магнітні або оптико-магнітні носії, які потім, в свою чергу, можуть використовуватись для введення інформації.

Для зберігання програм та інформації застосовують, головним чином магнітні, оптико-магнітні та оптичні диски, які дають можливість довільного доступу до даних і забезпечують високу швидкодію.

У зв'язку з бурхливим розвитком комп'ютерної техніки щороку зростає спектр їх різновидів. За призначенням комп'ютери можна умовно поділити на:

- побутові комп'ютери — власне ПК, що призначені для індивідуальної роботи в домашніх умовах;

- навчальні комп'ютери — призначені для використання в системі освіти, як вищої так і середньої. Основні вимоги таких комп'ютерів — надійність, достатня потужність та невисока ціна (можливість придбання більшої кількості однотипних комп'ютерів закладами освіти). Користувачі — учні, студенти, викладачі та ін.;

- професійні комп'ютери — робочі станції для роботи на виробництві, в офісах установ, які, як правило, об'єднані в локальну комп'ютерну мережу. Від „побутових” відрізняються більш високими показниками за всіма параметрами — продуктивністю, функціональними можливостями, якістю зображення на дисплеї та ін. Користувачі — службовці, оператори ВДТ;

- сервери — потужні комп'ютери, призначені для локальних та глобальних мереж. Вони виконують функції керування робочими станціями, зберігання значних масивів інформації та ін. Користувачі — менеджери, і адміністратори локальних комп'ютерних мереж, системні та прикладні програмісти для підтримки програмного забезпечення тощо;

- графічні станції — використовуються для роботи з графічними зображеннями, відео та анімацією. Володіють надзвичайно високими ресурсами за всіма основними параметрами.

- На сьогодні найбільш розповсюдженими є персональні комп'ютери (рис 7.1). В мінімальний базовий комплект ПК входять наступні блоки чи компоненти:

- системний блок, в якому зосереджені життєво важливі елементи комп'ютера;

- дисплей (монітор), який призначений для виведення (відображення) інформації;

- клавіатура, яка призначена для введення інформації в комп'ютер;

– графічний маніпулятор „миша”, який слугує для керування роботою програм шляхом вибору різних пунктів меню, виділення та „перетягування” об'єктів.

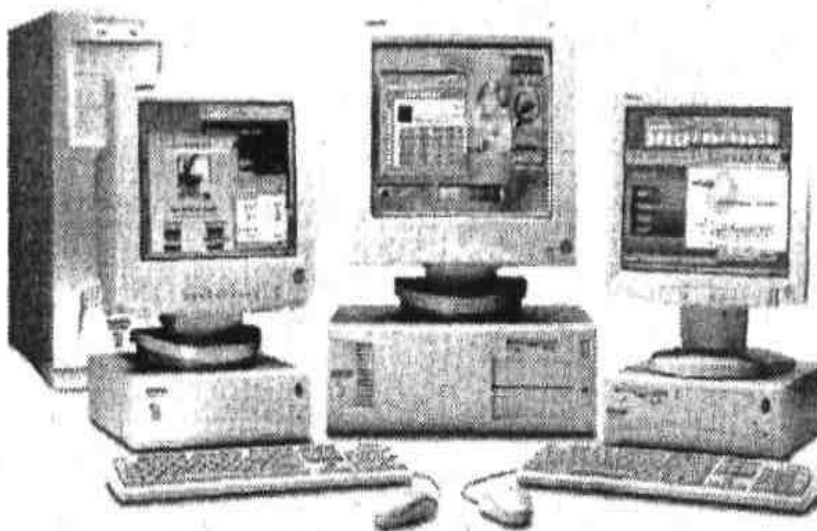


Рисунок 3.1 – Загальний вигляд ПК

При такому апаратному комплектуванні та при наявності відповідного програмного забезпечення вже можна повноцінно працювати за комп'ютером. Саме в такому комплектуванні комп'ютери, як правило, надходять у продаж.

3.1 План ліквідації аварій на виробничому об'єкті

Коротка характеристика аварій і катастроф.

Великі аварії і катастрофи на об'єктах можуть виникати в результаті стихійного лиха, а також порушення технології виробництва, правил експлуатації різних машин, обладнання і встановлених заходів безпеки. Їх дії подібні до стихійних лих.

Під аварією розуміють раптову зупинку роботи або порушення процесу виробництва на промисловому підприємстві, транспорті, інших об'єктах, що приводять до пошкодження або знищення матеріальних цінностей.

Під катастрофою розуміють раптове лихо; подію, що спричиняє за собою трагічні наслідки. Катастрофи супроводжуються руйнуванням будівель різних споруд, знищенням матеріальних цінностей і загибеллю людей.

Найбільш небезпечним наслідком великих аварій і катастроф є пожежі і вибухи. У ряді випадків, особливо на підприємствах нафтової, хімічної і газової промисловості, аварії викликають загазованість атмосфери, розлив нафтопродуктів, агресивних рідин і сильнодіючих отруйних речовин. Аварії і катастрофи можуть бути на залізничному, повітряному і водному транспорті, а також в результаті обвалення при будівництві і монтажі споруд і конструкцій різних об'єктів.

Основи використання формувань при стихійних лихах, великих аваріях і катастрофах. Для ліквідації наслідків, викликаних стихійними лихами, можуть притягуватися як формування загального призначення, так і формування служб ДО. В окремих випадках крім вказаних формувань можуть притягуватися військові частини ДО.

Основне завдання формувань при ліквідації наслідків стихійних лих, великих аварій і катастроф — спасіння людей і матеріальних цінностей. Характер і порядок дій формувань при виконанні цього завдання залежать від виду стихійного лиха, аварії або катастрофи, обстановки, що склалася, кількості і підготовленості сил цивільної оборони, що залучаються, пори року і доби, погодних умов і інших чинників.

Успіх дій формувань багато в чому залежить від своєчасної організації і проведення розвідки і обліку конкретних умов обстановки.

У районах стихійних лих розвідка визначає: межі осередку лиха і напряду його поширення, об'єкти і населені пункти, яким загрожує безпосередня небезпека, місця скупчення людей, шляхи підходу техніки до місць робіт, стан пошкоджених будівель і споруд, а також наявність в них уражених людей, місця аварій на комунально-енергетичних мережах, об'єм рятувальних і невідкладних аварійно-відновних робіт.

При крупних аваріях і катастрофах розвідка уточнює ступінь і об'єм руйнувань і можливість проведення робіт без засобів індивідуального захисту, можливість обвалення будівель і споруд, які можуть спричинити за собою збільшення розміру аварії або катастрофи, місця скупчення людей і ступінь загрози для їх життя, а також стан комунально-енергетичних мереж і транспортних комунікацій.

Розвідку ведуть розвідувальні групи і ланки. До складу розвідувальних формувань рекомендується включати фахівців, що знають розташування об'єкту і специфіку виробництва. Якщо в районі майбутніх дій можуть бути сильнодіючі отруйні речовини, то до складу розвідувальних формувань необхідно включати фахівців-хіміків і медичних працівників.

У зв'язку з раптовістю виникнення стихійних лих, великих аварій і катастроф сповіщення особового складу формувань, їх укомплектовування, створення угруповання проводяться в короткі терміни.

У перший ешелон угруповання сил зазвичай включаються формування об'єктів, де відбулися лиха, а в другій — формування сусідніх об'єктів (районів). Висунення формувань із районів збору в район дій здійснюється на максимально можливих швидкостях.

У районах стихійних лих і місцях великих аварій рятувальні роботи в першу чергу проводять з метою попередження виникнення катастрофічних наслідків, лих (аварій), запобігання виникненню вторинних причин, які можуть викликати загибель людей і матеріальних цінностей.

Командири формувань повинні постійно знати обстановку в районі робіт і відповідно до її зміни уточнювати або ставити нові завдання підрозділам.

Після виконання поставлених завдань формування виводяться в район постійного розквартирування.

Рятувальні і невідкладні аварійно-відновні роботи при ліквідації наслідків великих аварій і катастроф.

При землетрусах для проведення рятувальних і невідкладних аварійно-відновних робіт притягуються рятувальні, зведені загони (команди), загони

(команди) механізації робіт, аварійно-технічні команди, інші формування, які мають на озброєнні бульдозери, екскаватори, крани, механізований інструмент і засоби малої механізації.

При проведенні рятувальних робіт в осередку землетрусу перш за все витягують з-під завалів, із напівзруйнованих будівель людей, яким надають першу медичну допомогу, що горять; влаштовують в завалах проїзди; локалізують і усувають аварії на інженерних мережах, які загрожують життю людей або перешкоджають проведенню рятувальних робіт; обрушують або укріплюють конструкції будівель і споруд, що знаходяться в аварійному стані; обладнали пункти збору, що постраждали і медичні пункти; організовують водопостачання.

Послідовність і терміни виконання робіт встановлює начальник цивільної оборони об'єкту, що опинився в зоні землетрусу.

При повенях для проведення рятувальних робіт залучають рятувальні загони, команди і групи, а також відомчі спеціалізовані команди і підрозділи, озброєні плавзасобами, санітарні дружини і пости, гідрометеорологічні пости, розвідувальні групи і ланки, зведені загони (команди) механізації робіт, формування будівельних, ремонтно-будівельних організацій, охорони громадського порядку.

При великих аваріях і катастрофах організація робіт по ліквідації наслідків проводиться з урахуванням обстановки, що склалася після аварії або катастрофи, ступеня руйнування і пошкодження будівель і споруд, технологічного обладнання, агрегатів, характеру аварій на комунально-енергетичних мережах і пожеж, особливостей забудови території об'єкту і інших умов.

Роботи по організації ліквідації наслідків аварій і катастроф проводяться в стислі терміни: необхідно швидко врятувати людей, що знаходяться під уламками будівель, в завалених підвалах, і надати їм екстрену медичну допомогу, а також запобігти іншим катастрофічним наслідкам, пов'язаним із загибеллю людей і втратою великої кількості матеріальних цінностей.

Із виникненням аварії або катастрофи начальник цивільної оборони на підставі даних розвідки і особистого спостереження ухвалює рішення на ліквідацію наслідків і ставить завдання формуванням.

Начальники ділянок керують рятувальними і невідкладними аварійно-відновними роботами. Вони вказують командирам формувань найбільш доцільні прийоми і способи виконання робіт, визначають матеріально-технічне забезпечення, терміни закінчення робіт і представляють донесення про об'єм виконаних робіт, організовують живлення, зміну і відпочинок особового складу формувань.

Заходи щодо попередження великих аварій і катастроф. Великі виробничі аварії і катастрофи завдають великого збитку народному господарству, тому забезпечення безаварійної роботи має виключно велике державне значення. Сучасне промислове підприємство є складним інженерно-технічним комплексом. Успіх його роботи багато в чому залежить від стану інших підприємств галузі, об'єктів суміжних галузей, що забезпечують постачання по кооперації, а також від стану енергопостачання, транспортних комунікацій, зв'язку і т.п. Заходи щодо попередження аварій і катастроф є найбільш складними і трудомісткими. Вони представляють комплекс організаційних і інженерно-технічних заходів, направлених на виявлення і усунення причин аварій і катастроф, максимальне зниження можливих руйнувань і втрат у випадку, якщо ці причини повністю не вдається усунути, а також на створення сприятливих умов для організації і проведення рятувальних і невідкладних аварійно-відновних робіт.

Найбільш ефективним заходом є закладка в проекти новостворюваних об'єктів планувальних, технічних і технологічних рішень, які повинні максимально зменшити ймовірність виникнення аварій або значно понизити матеріальний збиток у випадку, якщо аварія відбудеться. Так, для зниження пожежної небезпеки передбачається зменшення питомої ваги матеріалів, що згорають. При проектуванні нових і реконструкції існуючих систем водопостачання враховується потреба у воді не тільки для виробничих цілей,

але і для випадку виникнення пожежі. Подібні рішення розробляються і по інших елементах виробництва. Враховуються вимоги охорони праці, техніка безпеки, правила експлуатації енергетичних установок, обладнання підйомного крану, місткостей під високим тиском і т.д. Таким чином, ці заходи розробляються і впроваджуються комплексно, із обхватом всіх питань, від яких залежить безаварійна робота об'єктів, з урахуванням їх виробничих і територіальних особливостей, із залученням всіх ланок керування виробничою діяльністю.

3.1 Висновок до третього розділу

В третьому розділі кваліфікаційної роботи описано основні питання з безпеки життєдіяльності, окрім цього розкриті мета та завдання БЖД. А також описані план ліквідації аварій на виробничому об'єкті, коротка характеристика аварій і катастроф, а також рятувальні і невідкладні аварійно-відновні роботи при ліквідації наслідків великих аварій і катастроф.

ВИСНОВКИ

В даній кваліфікаційній роботі розглянуто основні поняття та технології побудови програм для розпізнавання об'єктів на зображеннях, на основі цього була розроблена програма яка дозволяє розпізнавати котів на людей на зображенні.

При створенні програми було опрацьовано наступні задачі:

1. Було здійснено порівняння типових методів розпізнавання об'єктів на зображеннях.

2. Було проведено глибокий огляд методів розпізнавання для задачі виявлення на зображенні саме таких об'єктів як коти та люди.

3. Провести огляд типових етапів які застосовуються під час розпізнавання об'єктів на зображенні;

3. Здійснена розробка програми для розпізнавання об'єктів на зображенні;

4. Проведено порівняння різних підходів до розпізнавання на основі застосування моделей нейронних мереж (Yolo v8n, Yolo v8m, Yolo v8x);

5. Проведено тестування розробленої програми, та обгрунтовано використання Yolo v8x для задачі розпізнавання об'єктів на зображенні.

У розділі «Основи безпеки життєдіяльності, охорони праці» описані питання які стосуються безпеки життєдіяльності, зокрема нами розглянуті питання призначення та завдання БЖД, деякі варіанти ліквідації аварій на об'єктах.

ПЕРЕЛІК ДЖЕРЕЛ

1. Шевченко, С. І. Комп'ютерний зір: алгоритми розпізнавання та обробки зображень : монографія / С. І. Шевченко. Київ : Наукова думка, 2020. – 312 с.
2. Гончаренко, І. В. Методи машинного навчання для задач комп'ютерного зору : навч. посіб. / І. В. Гончаренко, М. О. Коваль. Львів : Львівська політехніка, 2021. 148 с.
3. Goodfellow, I. Deep Learning / Ian Goodfellow, Yoshua Bengio, Aaron Courville. Cambridge : MIT Press, 2016. 775 p.
4. Redmon, J. You Only Look Once: Unified, Real-Time Object Detection [Електронний ресурс] / Joseph Redmon, Santosh Divvala, Ross Girshick, Ali Farhadi // arXiv preprint. 2016. arXiv:1506.02640. Режим доступу: <https://arxiv.org/abs/1506.02640>
5. Bochkovski, A. YOLOv4: Optimal Speed and Accuracy of Object Detection [Електронний ресурс] / Alexey Bochkovski, Chien-Yao Wang, Hong-Yuan Mark Liao // arXiv preprint. 2020. arXiv:2004.10934. Режим доступу: <https://arxiv.org/abs/2004.10934>
6. Ultralytics. YOLOv8 Docs [Електронний ресурс]. 2023. Режим доступу: <https://docs.ultralytics.com/>
7. Соловійов, В. М. Обробка та аналіз зображень в Python / В. М. Соловійов. Харків : ХНУРЕ, 2022. – 120 с.
8. Опанасенко, Л. П. Основи глибинного навчання з застосуванням TensorFlow та Keras : навч. посіб. / Л. П. Опанасенко. Київ : КНЕУ, 2022. 198 с.
9. Конспект лекцій з дисципліни «Основи безпеки життєдіяльності».
10. Influence of deformation process in material at multiple cracking and fragmentation of nanocoating PO Maruschak, SV Panin, SR Ignatovich, IM Zakiev, IV Konovalenko, Theoretical and Applied Fracture Mechanics 57 (1), 43-48
11. Пам'ятка-інструктаж з питань охорони праці для студентів Тернопільського державного технічного університету імені Івана Пулюя. – Тернопіль.

12. Стручок, В. С., Стручок, О. С., & Мудра, Д. В. (2017). Навчальний посібник до написання розділу дипломного проекту та дипломної роботи "Безпека в надзвичайних ситуаціях" для студентів всіх спец. денної, заочної (дистанційної) та екстернатної форм навчання.

13. Стручок, В. С. (2022). Техноекологія та цивільна безпека. Частина «Цивільна безпека». Навчальний посібник.

14. Безпека в надзвичайних ситуаціях. Методичний посібник для здобувачів освітнього ступеня «магістр» всіх спеціальностей денної та заочної (дистанційної) форм навчання / укл.: Стручок В. С. Тернопіль: ФОП Паляниця В. А., 2022. 156 с

15. Automated algorithm for determining surface's oil capacity based on the analysis of the Abbot-Firestone diagram's parameters. Iaroslav Lytvynenko, Volodymyr Dzyura, Pavlo Maruschak CEUR Workshop Proceedings, 2024, 3896, pp. 74–79.

16. Development of an algorithm for identification of damage types on the surface of sheet metal Palianytsia, Y., Lytvynenko, I., Menou, A., Shymchuk, G., Dubchak, A. CEUR Workshop Proceedings, 2024, 3742, pp. 84–96.

17. Methodology of the Formation of Sports Matches Statistical Information Using Neural Networks Sorokivska, O., Lytvynenko, I., Sorokivskyi, O., Kozbur, H., Strutynska, I. CEUR Workshop Proceedings, 2023, 3628, pp. 389–403.

18. The Method of Computer Modeling of Heart Rhythm based on the Vector of Stationary and Stationary-related Random Sequences Onyskiv, P., Lytvynenko, I., Oleksandr, V., Shymchuk, G., Hotovych, V. CEUR Workshop, 2023, 3468, pp. 223–232.

19. Computer modeling of cardiac rhythm based on vector of stationary random sequences. Serhii Lupenko, Iaroslav Lytvynenko, Petro Onyskiv, Anatolii Lupenko, Oleksandr Volianyk, Olena Tsitsiura // Scientific Journal of TNTU. Tern.: TNTU, 2023. Vol 108. No 4. P. 131–143.

20. Krizhevsky, A. ImageNet Classification with Deep Convolutional Neural Networks / Alex Krizhevsky, Ilya Sutskever, Geoffrey E. Hinton // Communications of the ACM. 2017. Vol. 60, No. 6. P. 84–90. DOI: <https://doi.org/10.1145/3065386>
21. Ren, S. Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks / Shaoqing Ren, Kaiming He, Ross Girshick, Jian Sun // IEEE Transactions on Pattern Analysis and Machine Intelligence. 2017. Vol. 39, No. 6. – P. 1137–1149.
22. Liu, W. SSD: Single Shot MultiBox Detector / Wei Liu et al. // Computer Vision ECCV 2016. Springer, Cham, 2016. P. 21–37. DOI: https://doi.org/10.1007/978-3-319-46448-0_2
23. Lin, T.-Y. Microsoft COCO: Common Objects in Context / Tsung-Yi Lin et al. // European Conference on Computer Vision. Springer, Cham, 2014. P. 740–755. – DOI: https://doi.org/10.1007/978-3-319-10602-1_48
24. He, K. Deep Residual Learning for Image Recognition / Kaiming He, Xiangyu Zhang, Shaoqing Ren, Jian Sun // Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR). 2016. P. 770–778.
25. Russakovsky, O. ImageNet Large Scale Visual Recognition Challenge / Olga Russakovsky et al. // International Journal of Computer Vision. 2015. Vol. 115. P. 211–252. – DOI: <https://doi.org/10.1007/s11263-015-0816>
26. Khan, A. A survey of the recent architectures of deep convolutional neural networks / Asifullah Khan et al. // Artificial Intelligence Review. 2020. Vol. 53. – P. 5455–5516. – DOI: <https://doi.org/10.1007/s10462-020-09825-6>
27. Zisserman, A. Very Deep Convolutional Networks for Large-Scale Image Recognition / Karen Simonyan, Andrew Zisserman // arXiv preprint. 2014. – arXiv:1409.1556. Режим доступа: <https://arxiv.org/abs/1409.1556>
28. OpenCV. Open Source Computer Vision Library [Электронный ресурс]. – Режим доступа: <https://opencv.org/> Дата звернення: 18.06.2025.
29. Tan, M. EfficientDet: Scalable and Efficient Object Detection / Mingxing Tan, Ruoming Pang, Quoc V. Le // Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. 2020. P. 10781–10790.

ДОДАТКИ

Приклад коду програми для розпізнавання об'єктів на зображенні (YOLO v8n)

```
import cv2
import numpy as np
import matplotlib.pyplot as plt
from ultralytics import YOLO
import os

# Створюємо директорії для збереження результатів
os.makedirs("output", exist_ok=True)

# Завантажуємо модель YOLOv8 (tiny модель, швидка для тесту)
model = YOLO("yolov8n.pt")

# Завантажуємо зображення
image_path = "input/photo.jpg" # <-- сюди клади своє зображення
img_bgr = cv2.imread(image_path)
img_rgb = cv2.cvtColor(img_bgr, cv2.COLOR_BGR2RGB)

# Зберігаємо оригінал
plt.imsave("output/step1_original.jpg", img_rgb)

# Етап 2: Ресайз (імітація підготовки для моделі)
resized = cv2.resize(img_rgb, (640, 640))
plt.imsave("output/step2_resized.jpg", resized)

# Етап 3: Передбачення моделі
results = model(resized)

# Етап 4: Виведення heatmap активності (імітація)
heatmap = cv2.applyColorMap(cv2.cvtColor(resized,
cv2.COLOR_RGB2GRAY), cv2.COLORMAP_JET)
plt.imsave("output/step3_heatmap.jpg", heatmap)

# Етап 5: Первинні бокси (імітація)
result_img_initial = resized.copy()
for r in results:
    for box in r.bboxes:
        xyxy = box.xyxy.cpu().numpy()[0].astype(int)
        cv2.rectangle(result_img_initial, (xyxy[0], xyxy[1]), (xyxy[2], xyxy[3]), (0,
255, 255), 2)
```

```

plt.imshow("output/step4_initial_boxes.jpg", result_img_initial)

# Етап 6: Фінальні бокси (остаточні)
result_img_final = resized.copy()
for r in results:
    for box in r.bboxes:
        xyxy = box.xyxy.cpu().numpy()[0].astype(int)
        cls_id = int(box.cls.cpu().numpy())
        conf = float(box.conf.cpu().numpy())
        label = f"{model.names[cls_id]} {conf:.2f}"
        cv2.rectangle(result_img_final, (xyxy[0], xyxy[1]), (xyxy[2], xyxy[3]), (0,
255, 0), 2)
        cv2.putText(result_img_final, label, (xyxy[0], xyxy[1]-10),
cv2.FONT_HERSHEY_SIMPLEX, 0.6, (255,255,255), 2)
plt.imshow("output/step5_final_boxes.jpg", result_img_final)

# Друкуємо таблицю результатів
print("\nРозпізнані об'єкти:")
print("{:10} {:10} {:30}".format("Class", "Confidence", "Coordinates"))
for r in results:
    for box in r.bboxes:
        cls_id = int(box.cls.cpu().numpy())
        conf = float(box.conf.cpu().numpy())
        xyxy = box.xyxy.cpu().numpy()[0].astype(int)
        print(f"{model.names[cls_id]:10} {conf:.2f} {xyxy}")

```

Приклад коду програми для розпізнавання об'єктів на зображенні (YOLO v8m)

```
import cv2
import numpy as np
import matplotlib.pyplot as plt
from ultralytics import YOLO
import os

# Створюємо директорії для збереження результатів
os.makedirs("output", exist_ok=True)

# Завантажуємо модель YOLOv8 (середня модель для кращої точності)
model = YOLO("yolov8m.pt") # використаємо більш точну модель

# Завантажуємо зображення
image_path = "input/photo.jpg" # <-- сюди клади своє зображення
img_bgr = cv2.imread(image_path)
img_rgb = cv2.cvtColor(img_bgr, cv2.COLOR_BGR2RGB)

# Зберігаємо оригінал
plt.imsave("output/step1_original.jpg", img_rgb)

# Етап 2: Ресайз (імітація підготовки для моделі)
resized = cv2.resize(img_rgb, (640, 640))
plt.imsave("output/step2_resized.jpg", resized)

# Етап 3: Передбачення моделі з вищим порогом впевненості для котів
results = model(resized, conf=0.25)

# Етап 4: Виведення heatmap активності (імітація)
heatmap = cv2.applyColorMap(cv2.cvtColor(resized,
cv2.COLOR_RGB2GRAY), cv2.COLORMAP_JET)
plt.imsave("output/step3_heatmap.jpg", heatmap)

# Етап 5: Первинні бокси (імітація)
result_img_initial = resized.copy()
for r in results:
    for box in r.bboxes:
        xyxy = box.xyxy.cpu().numpy()[0].astype(int)
```

```

    cv2.rectangle(result_img_initial, (xyxy[0], xyxy[1]), (xyxy[2], xyxy[3]), (0,
255, 255), 2)
plt.imshow("output/step4_initial_boxes.jpg", result_img_initial)

# Етап 6: Фінальні бокси (остаточні з фільтром по класу кота)
result_img_final = resized.copy()
for r in results:
    for box in r.bboxes:
        xyxy = box.xyxy.cpu().numpy()[0].astype(int)
        cls_id = int(box.cls.cpu().numpy())
        conf = float(box.conf.cpu().numpy())
        label = f"{model.names[cls_id]} {conf:.2f}"
        if model.names[cls_id] == 'cat' and conf < 0.40:
            continue # пропускаємо слабкі виявлення кота
        cv2.rectangle(result_img_final, (xyxy[0], xyxy[1]), (xyxy[2], xyxy[3]), (0,
255, 0), 2)
        cv2.putText(result_img_final, label, (xyxy[0], xyxy[1]-10),
cv2.FONT_HERSHEY_SIMPLEX, 0.6, (255,255,255), 2)
plt.imshow("output/step5_final_boxes.jpg", result_img_final)

# Друкуємо таблицю результатів
print("\nРозпізнані об'єкти:")
print("{:10} {:10} {:30}".format("Class", "Confidence", "Coordinates"))
for r in results:
    for box in r.bboxes:
        cls_id = int(box.cls.cpu().numpy())
        conf = float(box.conf.cpu().numpy())
        xyxy = box.xyxy.cpu().numpy()[0].astype(int)
        print(f"{model.names[cls_id]:10} {conf:.2f} {xyxy}")

# Малюємо результати
output_image = draw_bbox(image, bbox, label, conf)

# Виводимо результат
cv2.imshow("Object Detection", output_image)
cv2.waitKey()
cv2.destroyAllWindows()

```

Приклад коду програми для розпізнавання об'єктів на зображенні (YOLO v8x)

```
import cv2
import numpy as np
import matplotlib.pyplot as plt
from ultralytics import YOLO
import os

# Створюємо директорії для збереження результатів
os.makedirs("output", exist_ok=True)

# Завантажуємо більш точну модель YOLOv8x
model = YOLO("yolov8x.pt")

# Завантажуємо зображення
image_path = "input/photo.jpg" # <-- сюди клади своє зображення
img_bgr = cv2.imread(image_path)
img_rgb = cv2.cvtColor(img_bgr, cv2.COLOR_BGR2RGB)

# Зберігаємо оригінал
plt.imshow("output/step1_original.jpg", img_rgb)

# Етап 2: Ресайз (імітація підготовки для моделі)
resized = cv2.resize(img_rgb, (640, 640))
plt.imshow("output/step2_resized.jpg", resized)

# Етап 3: Передбачення моделі з вищим порогом впевненості для кращої якості
results = model(resized, conf=0.25)

# Етап 4: Виведення heatmap активності (імітація)
heatmap = cv2.applyColorMap(cv2.cvtColor(resized, cv2.COLOR_RGB2GRAY), cv2.COLORMAP_JET)
plt.imshow("output/step3_heatmap.jpg", heatmap)

# Етап 5: Первинні бокси (імітація)
result_img_initial = resized.copy()
for r in results:
    for box in r.bboxes:
        xyxy = box.xyxy.cpu().numpy()[0].astype(int)
```

```

cv2.rectangle(result_img_initial, (xyxy[0], xyxy[1]), (xyxy[2], xyxy[3]), (0,
255, 255), 2)
plt.imshow("output/step4_initial_boxes.jpg", result_img_initial)

```

Етап 6: Фінальні бокси (остаточні з фільтром по класу kota)

```
result_img_final = resized.copy()
```

```
for r in results:
```

```
    for box in r.bboxes:
```

```
        xyxy = box.xyxy.cpu().numpy()[0].astype(int)
```

```
        cls_id = int(box.cls.cpu().numpy())
```

```
        conf = float(box.conf.cpu().numpy())
```

```
        label = f"{model.names[cls_id]} {conf:.2f}"
```

```
        if model.names[cls_id] == 'cat' and conf < 0.40:
```

```
            continue # пропускаємо слабкі виявлення кота
```

```
        cv2.rectangle(result_img_final, (xyxy[0], xyxy[1]), (xyxy[2], xyxy[3]), (0,
255, 0), 2)
```

```
        cv2.putText(result_img_final, label, (xyxy[0], xyxy[1]-10),
cv2.FONT_HERSHEY_SIMPLEX, 0.6, (255,255,255), 2)
```

```
plt.imshow("output/step5_final_boxes.jpg", result_img_final)
```

Друкуємо таблицю результатів

```
print("\nРозпізнані об'єкти:")
```

```
print("{:10} {:10} {:30}".format("Class", "Confidence", "Coordinates"))
```

```
for r in results:
```

```
    for box in r.bboxes:
```

```
        cls_id = int(box.cls.cpu().numpy())
```

```
        conf = float(box.conf.cpu().numpy())
```

```
        xyxy = box.xyxy.cpu().numpy()[0].astype(int)
```

```
        print(f"{model.names[cls_id]:10} {conf:.2f} {xyxy}")
```