

Міністерство освіти і науки України
Харківський національний університет радіоелектроніки

Факультет _____ Комп'ютерних наук _____
(повна назва)
Кафедра _____ Штучного інтелекту _____
(повна назва)

КВАЛІФІКАЦІЙНА РОБОТА
Пояснювальна записка

рівень вищої освіти _____ перший (бакалаврський) _____

Інтелектуальна система детекції аномальних станів ігрового середовища для
превентивного забезпечення безпеки дитини

(тема)

Виконав:
здобувач _____ четвертого _____ року навчання,
групи _____ ІТШ-22-4 _____

_____ Ольга СТАХЕВИЧ _____
(власне ім'я, прізвище)

Спеціальність 122 Комп'ютерні науки _____

(код і повна назва спеціальності)

Тип програми _____ освітньо-професійна _____

Освітня програма _____ Штучний інтелект _____

(повна назва освітньої програми)

Керівник ст. викл. Олена ГРИНЬОВА _____
(посада, власне ім'я, прізвище)

Допускається до захисту

Завідувач кафедри ШІ

(підпис)

_____ Лариса ЧАЛА _____
(власне ім'я, прізвище)

2026 р.

Харківський національний університет радіоелектроніки

Факультет _____ Комп'ютерних наук _____

Кафедра _____ Штучного інтелекту _____

Рівень вищої освіти _____ перший (бакалаврський) _____

Спеціальність _____ 122 Комп'ютерні науки _____

(код і повна назва)

Тип програми _____ освітньо-професійна _____

Освітня програма _____ Штучний інтелект _____

(повна назва)

ЗАТВЕРДЖУЮ:

Зав. кафедри _____

(підпис)

« _____ » _____ 20 ____ р.

ЗАВДАННЯ НА КВАЛІФІКАЦІЙНУ РОБОТУ

здобувачеві _____ Стахевич Ользі Віталіївні _____

(прізвище, ім'я, по батькові)

1. Тема роботи _____ Інтелектуальна система детекції аномальних станів ігрового середовища для превентивного забезпечення безпеки дитини _____

затверджена наказом університету від 5 червня 2026 р. № 488Ст

2. Термін подання здобувачем роботи до екзаменаційної комісії 8 липня 2026 р.

3. Вихідні дані до роботи Матеріали передатестаційної практики; тема кваліфікаційної роботи; зображення ігрового середовища дитини; набір даних для детекції об'єктів класів baby, knife, medicine, scissors, socket; модель YOLO11s; середовище Google Colab; мова програмування Python; бібліотеки Ultralytics YOLO, OpenCV, PyTorch; фреймворк Streamlit; методи комп'ютерного зору, глибинного навчання та оцінювання просторової близькості об'єктів.

4. Перелік питань, що потрібно опрацювати в роботі _____

1) Аналіз предметної галузі та постановка задачі _____

2) Постановка задачі та проєктування системи _____

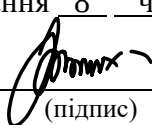
3) Розроблення та навчання моделі детекції об'єктів _____

4) Експериментальні дослідження та аналіз результатів _____

КАЛЕНДАРНИЙ ПЛАН

№	Назва етапів роботи	Строк / термін виконання етапів роботи	Примітка
1	Отримання завдання на кваліфікаційну роботу	08.06.2026	виконано
2	Аналіз предметної галузі та постановка задачі	08.06.2026	виконано
3	Постановка задачі дослідження та проектування інтелектуальної системи	10.06.2026	виконано
4	Розроблення та навчання моделі детекції об'єктів	15.06.2026	виконано
5	Експериментальні дослідження та аналіз результатів	17.06.2026	виконано
6	Написання пояснювальної записки	20.06.2026	виконано
7	Перевірка на академічний плагіат	21.06.2026	виконано
8	Нормоконтроль	24.06.2026	виконано
9	Підготовка презентації та доповіді	24.06.2026	виконано
10	Попередній захист	26.06.2026	виконано
11	Рецензування	27.06.2026	виконано
12	Захист перед ЕК	08.07.2026	

Дата видачі завдання 8 червня 20 26 р.

Здобувач 
(підпис)

Керівник роботи 
(підпис)

ст. викл. Олена ГРИНЬОВА
(посада, власне ім'я, прізвище)

РЕФЕРАТ

Пояснювальна записка: 90 с., 8 рис., 6 табл., 1 дод., 26 джерел.

БЕЗПЕКА ДИТИНИ, ДЕТЕКЦІЯ АНОМАЛЬНИХ СТАНІВ, ІГРОВЕ СЕРЕДОВИЩЕ, ІНТЕЛЕКТУАЛЬНА СИСТЕМА, КОМП'ЮТЕРНИЙ ЗІР, YOLO.

Об'єкт дослідження – інтелектуальні системи моніторингу безпеки дитячого ігрового середовища, що базуються на методах комп'ютерного зору та глибинного навчання.

Предмет дослідження – алгоритмічні та програмні методи автоматичного виявлення дитини й потенційно небезпечних предметів у зображеннях, а також оцінювання їх просторової близькості для виявлення ризикових ситуацій.

Мета роботи – розробити концепцію інтелектуальної системи детекції аномальних станів ігрового середовища для превентивного забезпечення безпеки дитини.

Методи дослідження – аналіз предметної області, методи комп'ютерного зору, глибинного навчання, детекції об'єктів на основі нейронних мереж, геометричний аналіз просторової близькості та порогове формування попереджень.

Протягом написання роботи розглянуто актуальність проблеми безпеки дитини в домашньому ігровому середовищі, основні джерела небезпеки, сучасні підходи до моніторингу безпеки, принципи роботи систем комп'ютерного зору та нейронних мереж, особливості архітектури YOLO11, а також методи оцінювання аномальних станів і формування попереджень про небезпечні ситуації.

ABSTRACT

Bachelor's thesis contains: 90 pp., 8 fig., 6 tabl., 1 ann., 26 references.

ANOMALY DETECTION, CHILD SAFETY, COMPUTER VISION,
INTELLIGENT SYSTEM, PLAY ENVIRONMENT, YOLO.

Research Object – intelligent systems for monitoring the safety of children's play environments based on computer vision and deep learning methods.

Research Subject – algorithmic and software methods for automatically detecting children and potentially dangerous objects in images, as well as assessing their spatial proximity to identify risky situations.

The aim of the work is to develop a concept for an intelligent system for detecting abnormal conditions in the play environment to ensure the child's safety proactively.

Research methods include domain analysis, computer vision techniques, deep learning, object detection based on neural networks, geometric analysis of spatial proximity, and threshold-based warning generation.

Throughout the thesis, the relevance of child safety in the home play environment, the main sources of danger, modern approaches to safety monitoring, the principles of computer vision and neural network systems, the features of the YOLO11 architecture, as well as methods for assessing abnormal conditions and generating warnings about dangerous situations are examined.

ЗМІСТ

Перелік умовних позначень, символів, одиниць, скорочень і термінів	8
Вступ.....	9
1 Аналіз предметної галузі та постановка задачі.....	11
1.1 Опис предметної галузі	11
1.1.1 Особливості ігрового середовища дитини	11
1.1.2 Основні джерела небезпеки та проблеми забезпечення безпеки дітей.....	13
1.2 Аналіз сучасних методів моніторингу безпеки	14
1.3 Основи комп'ютерного зору та нейронних мереж.....	16
1.3.1 Принципи роботи систем комп'ютерного зору	17
1.3.2 Методи виявлення та класифікації об'єктів.....	19
1.3.3 Застосування нейронних мереж у задачах безпеки.....	20
1.3.4 Архітектура yolov та її місце в детекції об'єктів.....	22
1.3.5 Особливості та переваги yolov11	25
1.4 Методи оцінювання аномальних станів	27
1.4.1 Оцінювання просторової близькості об'єктів.....	27
1.4.2 Формування попереджень про небезпечні ситуації.....	30
2 Постановка задачі та проектування системи.....	32
2.1 Формування вимог до інтелектуальної системи.....	32
2.2 Вибір підходу до реалізації системи та очікувані результати.....	35
2.3 Загальна архітектура інтелектуальної системи.....	38
2.4 Визначення класів об'єктів для детекції.....	40
2.5 Розроблення алгоритму оцінювання рівня небезпеки	42
2.6 Розроблення алгоритму виявлення аномальних станів	44
2.7 Проектування структури користувацького інтерфейсу	46
3 Розроблення та навчання моделі детекції об'єктів.....	48
3.1 Формування та підготовка набору даних	48
3.2 Аналіз структури датасету	51

3.3 Розподіл даних на навчальну, валідаційну та тестову вибірки	53
3.4 Аугментація даних та її вплив на процес навчання	54
3.5 Налаштування середовища навчання.....	57
3.6 Навчання моделі yolo11s	59
3.7 Аналіз результатів навчання	61
3.8 Реалізація програмного забезпечення засобами python	62
3.9 Розроблення вебзастосунку на основі streamlit.....	65
4 Експериментальні дослідження та аналіз результатів	67
4.1 Методика проведення експериментальних досліджень.....	67
4.2 Характеристика тестової вибірки	68
4.3 Оцінювання ефективності моделі за метриками precision, recall, map50 та map50-95	69
4.4 Аналіз результатів детекції дитини та небезпечних об'єктів.....	71
4.5 Аналіз типових помилок системи	74
4.6 Аналіз динаміки навчання моделі протягом 40 епох	76
4.7 Демонстрація роботи системи на тестових зображеннях	77
4.8 Оцінювання практичної придатності розробленої системи	79
Висновки	83
Перелік джерел посилання	86
Додаток А Відомість кваліфікаційної роботи	90

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ, СКОРОЧЕНЬ І ТЕРМІНІВ

ІІІ – штучний інтелект;

AI – Artificial Intelligence – штучний інтелект;

Bounding box – обмежувальна рамка об'єкта на зображенні;

CNN – Convolutional Neural Network – згорткова нейронна мережа;

CPU – Central Processing Unit – центральний процесор;

CV – Computer Vision – комп'ютерний зір;

GPU – Graphics Processing Unit – графічний процесор;

IoU – Intersection over Union – показник перетину обмежувальних рамок;

mAP – mean Average Precision – середня точність детекції об'єктів;

mAP50 – середня точність детекції за порогу IoU 0,50;

mAP50-95 – середня точність детекції за набором порогів IoU від 0,50 до 0,95;

OpenCV – бібліотека комп'ютерного зору для оброблення зображень;

Precision – точність спрацювань моделі;

Recall – повнота виявлення об'єктів;

Roboflow – платформа для підготовки та експорту наборів даних комп'ютерного зору;

Streamlit – фреймворк для створення вебзастосунків мовою Python;

YOLO – You Only Look Once – сімейство моделей одноетапної детекції об'єктів.

ВСТУП

Сучасний етап розвитку інформаційних технологій характеризується активним впровадженням інтелектуальних систем у різні сфери повсякденного життя. Особливої уваги набувають рішення, що спрямовані на підвищення рівня безпеки людини, автоматизацію контролю та своєчасне виявлення ризиків. Завдяки застосуванню методів штучного інтелекту, комп'ютерного зору та обробки зображень стає можливим створення систем, здатних аналізувати навколишнє середовище, розпізнавати об'єкти та реагувати на зміни в сцені без безпосередньої участі людини.

У побутових умовах особливої актуальності набувають засоби контролю, які можуть забезпечити додатковий рівень захисту для дітей. Дитина, перебуваючи в ігровому середовищі, часто взаємодіє з великою кількістю предметів, частина з яких може становити потенційну небезпеку. Через обмежений життєвий досвід і природну допитливість діти не завжди здатні самостійно оцінити ризик, тому питання своєчасного виявлення небезпечних ситуацій є важливим як у домашньому, так і в освітньому середовищі.

У цьому контексті перспективним є застосування інтелектуальних систем, які можуть автоматично виявляти об'єкти в кадрі, визначати їхнє розташування та оцінювати просторові взаємозв'язки між ними. Такі системи дають змогу не лише фіксувати присутність потенційно небезпечних предметів, а й виявляти випадки, коли дитина наближається до них на критичну відстань. Саме тому дослідження методів автоматичної детекції та аналізу аномальних станів середовища є важливим напрямом сучасної прикладної інформатики.

Актуальність теми також зумовлена необхідністю створення рішень, які можуть працювати в режимі реального часу, бути достатньо точними та придатними для використання в умовах змінного освітлення, часткових перекриттів і складного побутового фону. Це обумовлює доцільність

використання сучасних нейронних мереж, зокрема архітектур сімейства YOLO, які забезпечують високу швидкість обробки зображень і можуть бути адаптовані до спеціалізованих задач детекції.

У межах цієї роботи особливу увагу приділено не лише факту виявлення небезпечних предметів, а й аналізу їхнього просторового розташування відносно дитини. Такий підхід є важливим, оскільки один і той самий об'єкт може мати різний рівень небезпеки залежно від ситуації: предмет, що знаходиться на значній відстані, не потребує негайного реагування, тоді як його близьке розташування до дитини може свідчити про потенційно ризиковий стан. Саме тому поєднання детекції об'єктів із подальшим оцінюванням відстані між ними дозволяє зробити систему більш інформативною та наближеною до практичних умов використання.

Отже, тема роботи є актуальною, оскільки поєднує практичну потребу в підвищенні безпеки дитини з можливостями сучасних інтелектуальних технологій. Розроблення системи, здатної автоматично виявляти аномальні стани ігрового середовища та попереджати про можливу небезпеку, має важливе значення для створення більш надійних засобів превентивного контролю.

1 АНАЛІЗ ПРЕДМЕТНОЇ ГАЛУЗІ ТА ПОСТАНОВКА ЗАДАЧІ

1.1 Опис предметної галузі

Питання безпеки дитини в домашньому середовищі є одним із пріоритетних напрямів сучасних досліджень у сфері охорони здоров'я, інженерії безпеки та інтелектуальних систем моніторингу. Домашній простір, який традиційно сприймається як відносно безпечне середовище, на практиці містить значну кількість факторів ризику, пов'язаних із побутовими предметами, електроприладами, меблями, дрібними деталями іграшок, ліками та іншими об'єктами, здатними спричинити травмування або отруєння дитини [1]. Особливо вразливими є діти раннього та дошкільного віку, оскільки вони ще не мають достатньо сформованих навичок оцінювання небезпеки, а також демонструють високу рухову активність, цікавість до навколишніх предметів і схильність до експериментування [2].

У контексті інтелектуальних систем безпеки предметна область охоплює не лише виявлення небезпечних об'єктів, а й аналіз взаємного розташування дитини та потенційних джерел ризику, оцінювання їхньої близькості в межах зображення, формування попереджень і підтримку прийняття рішень дорослими. Саме тому задача забезпечення безпеки дітей в ігровому середовищі є міждисциплінарною: вона поєднує знання з дитячої гігієни, теорії ризику, комп'ютерного зору, машинного навчання та обробки зображень.

1.1.1 Особливості ігрового середовища дитини

Ігрове середовище дитини є динамічною та змінною частиною житлового простору, яка не має фіксованої структури. На відміну від спеціалізованих контрольованих зон, таких як лабораторії або виробничі

приміщення, дитяча кімната або ігрова зона в квартирі формується залежно від побутових умов, наявності меблів, іграшок, техніки та звичок дорослих. У такому середовищі об'єкти постійно переміщуються, частково перекривають один одного, змінюють видимість і можуть тимчасово опинятися в зоні досяжності дитини.

Характерною особливістю ігрового середовища є поєднання безпечних і потенційно небезпечних об'єктів у межах однієї сцени. До безпечних елементів належать іграшки, м'які предмети, дитячі меблі та предмети догляду, тоді як до небезпечних можуть належати ліки, гострі предмети, електричні розетки, кабелі, скляні або крихкі предмети, засоби побутової хімії, дрібні деталі та інші об'єкти, що створюють ризик травмування або отруєння.

Ще однією важливою особливістю є те, що в ігровому середовищі поведінка дитини може бути непередбачуваною. Дитина може швидко змінювати положення, нахилитися, тягнутися до предметів, переносити іграшки або взаємодіяти з об'єктами, які зовні здаються безпечними, але в конкретній ситуації можуть становити ризик. Це означає, що для оцінювання безпеки недостатньо лише зафіксувати наявність небезпечного предмета; необхідно враховувати його розташування відносно дитини, масштаб сцени та контекст розміщення об'єктів.

У практичному сенсі домашнє ігрове середовище часто має складні умови для автоматичного аналізу: змішане освітлення, тіні, часткові перекриття, фрагментарну видимість об'єктів, змінний фон і різні ракурси камери. Саме тому воно є зручним, але складним об'єктом для застосування методів комп'ютерного зору. Для інтелектуальної системи безпеки це означає необхідність працювати не лише з категоріями об'єктів, а й із їхніми геометричними характеристиками на зображенні, такими як координати, площа, центр рамки та піксельна відстань між об'єктами.

1.1.2 Основні джерела небезпеки та проблеми забезпечення безпеки дітей

Основні джерела небезпеки в ігровому середовищі дитини можна умовно поділити на такі групи [3]:

- механічні небезпеки – гострі предмети, зокрема ножі та ножиці, дрібні деталі іграшок, а також інші предмети, які можуть спричинити поранення або удушення;

- хімічні небезпеки – ліки, побутова хімія, спиртовмісні речовини та інші токсичні матеріали, доступ до яких може становити ризик для здоров'я дитини;

- електричні небезпеки – розетки, подовжувачі, зарядні пристрої, оголені дроти або несправна техніка, контакт із якими може призвести до ураження електричним струмом;

- термічні небезпеки – гарячі поверхні, обігрівачі, праски, чайники, свічки та інші джерела тепла, які можуть спричинити опіки;

- побутові небезпеки – меблі, що можуть упасти, ковзкі поверхні, відкриті вікна, сходи або предмети, які дитина може перекинути під час гри.

Проблема забезпечення безпеки дитини полягає в тому, що небезпека в домашньому середовищі часто має латентний характер. Більшість небезпечних об'єктів не становлять загрози самі по собі, але можуть стати небезпечними у разі наближення дитини, дотику або спроби використання під час гри. Тому важливим є не лише виявлення потенційно небезпечного предмета, а й оцінювання його просторового положення відносно дитини. Саме це створює підґрунтя для використання підходів комп'ютерного зору, у яких аналізується взаємне розташування дитини та небезпечних об'єктів, після чого за заданими правилами визначається рівень ризику ситуації.

Існує також низка практичних труднощів, які ускладнюють забезпечення безпеки дітей у побутових умовах. Насамперед це людський фактор, оскільки батьки або опікуни не можуть постійно контролювати

поведінку дитини, особливо під час виконання інших побутових або робочих завдань. Навіть короткочасна втрата уваги може призвести до небажаної ситуації. Крім того, традиційні камери відеоспостереження не завжди розв'язують цю проблему, оскільки переважно лише фіксують зображення або відео, але не забезпечують автоматичного аналізу потенційно небезпечних об'єктів і їхнього розташування відносно дитини.

Додатковою проблемою є недостатня адаптованість існуючих рішень до умов житлового простору. Комерційні системи безпеки найчастіше орієнтовані на контроль входів, периметрів або загального руху, тоді як особливості дитячого середовища залишаються поза увагою. Для ефективного контролю потрібні системи, здатні не лише розпізнавати об'єкти на зображенні, а й оцінювати потенційний ризик їхнього розташування відносно дитини. Це обґрунтовує актуальність застосування методів комп'ютерного зору та глибинного навчання для створення інтелектуальних систем моніторингу безпеки.

1.2 Аналіз сучасних методів моніторингу безпеки

Забезпечення безпеки дитини в домашньому або навчальному середовищі вимагає поєднання організаційних, технічних та інтелектуальних заходів. Традиційні методи, зокрема постійний нагляд дорослих, фізичні обмежувачі, правила поведінки та інструктажі, є базовими, однак їхня ефективність значною мірою залежить від людського фактора. У закладах освіти додатково застосовують регламенти евакуації, інструкції з охорони праці та регулярні перевірки приміщень. Попри практичну корисність, такі заходи не забезпечують безперервного автоматизованого аналізу та не завжди дозволяють швидко реагувати на зміну обстановки [4].

Інтелектуальні системи контролю розвиваються як відповідь на зазначені обмеження. Вони поєднують датчики, камери, алгоритми

комп'ютерного зору та машинного навчання для автоматичного виявлення подій, пов'язаних із безпекою: наявності людини в зоні ризику, відкриття дверей, диму, аномальної температури або небезпечних взаємозв'язків між об'єктами. На відміну від звичайних камер спостереження, такі системи не лише фіксують зображення, а й інтерпретують візуальну інформацію, розпізнають об'єкти, визначають їхню позицію та формують контекстно-залежні попередження. У контексті безпеки дитини це дає змогу автоматично виявляти ситуації, коли небезпечний предмет перебуває в зоні досяжності або розташований надто близько до дитини.

Попри активний розвиток технологій, існуючі системи мають суттєві обмеження.

По-перше, традиційні засоби покладаються на увагу людини, яка не гарантує безперервного нагляду.

По-друге, більшість комерційних рішень орієнтовані на загальний контроль доступу або виявлення руху і не враховують специфіку дитячого середовища. Вони рідко оцінюють, чи становить конкретний об'єкт ризик саме для дитини, та не аналізують просторові відносини між дитиною і джерелом небезпеки.

По-третє, поширені хибні спрацювання, зумовлені реакцією на тварин, зміну освітлення або випадкові предмети, знижують довіру користувачів і можуть призводити до ігнорування важливих повідомлень. Нарешті, домашні умови зі змінним фоном, частковими перекриттями та різним освітленням ускладнюють застосування простих алгоритмів без використання методів глибинного навчання.

Отже, існуючі підходи або надто залежать від людини, або недостатньо адаптовані до особливостей дитячого середовища. Це обґрунтовує необхідність розробки спеціалізованого підходу, який поєднує детекцію об'єктів з оцінюванням їхнього просторового розташування відносно дитини та контекстною класифікацією ризику.

1.3 Основи комп'ютерного зору та нейронних мереж

Комп'ютерний зір є одним із найбільш динамічних напрямів штучного інтелекту, який спрямований на автоматичне отримання, аналіз та інтерпретацію візуальної інформації, поданої у вигляді зображень або окремих кадрів. Основне завдання комп'ютерного зору полягає не лише у фіксації сцени, а й у перетворенні необроблених піксельних даних у структуроване подання, придатне для подальшої класифікації, детекції, сегментації, трекінгу та прийняття рішень. У сучасних системах безпеки комп'ютерний зір використовується для розпізнавання людей, виявлення небезпечних об'єктів, аналізу ситуацій у кадрі та оцінювання взаємного розташування об'єктів [5].

Зростання значення комп'ютерного зору пов'язане з розвитком алгоритмів машинного та глибокого навчання, які значно підвищили якість розпізнавання зображень у порівнянні з класичними методами. Якщо раніше для аналізу візуальних даних переважно застосовувалися ручно спроектовані ознаки, то сучасні моделі навчаються автоматично виявляти інформативні представлення безпосередньо з даних. Це дозволяє працювати зі складними сценами, де присутні часткові перекриття, різні масштаби об'єктів, шум, зміни освітлення та варіативність фонового середовища.

Для задачі моніторингу безпеки дитини комп'ютерний зір має особливу цінність, оскільки дає змогу автоматизувати виявлення дитини та потенційно небезпечних предметів у домашньому просторі. Такі системи можуть аналізувати зображення або окремий кадр, визначати координати об'єктів, оцінювати їхнє взаємне розташування та формувати попередження у разі виявлення потенційно ризикової ситуації. Саме тому подальше дослідження принципів комп'ютерного зору та архітектур нейронних мереж є необхідним для обґрунтування вибору методів, що застосовуються у даній роботі.

1.3.1 Принципи роботи систем комп'ютерного зору

Робота системи комп'ютерного зору починається з отримання вхідного зображення або окремого відеокадру. З технічної точки зору зображення являє собою двовимірну або тривимірну матрицю пікселів, де кожен елемент містить інформацію про яскравість або колір. У разі кольорових зображень кожен піксель подається у вигляді значень кольорових каналів, найчастіше RGB. Перед подальшою обробкою зображення може проходити етапи нормалізації, масштабування, вирівнювання яскравості, фільтрації шумів або зміни розміру. Такі операції спрямовані на уніфікацію вхідних даних і підвищення стійкості алгоритму до варіативності сцени.

У традиційних алгоритмах комп'ютерного зору ключове значення мали етапи ручного виділення ознак. До таких ознак належали контури, кути, текстури, градієнти, гістограми орієнтованих градієнтів (HOG) та інші характеристики, які допомагали описувати структуру зображення. Однак ефективність таких методів суттєво залежала від якості вибраних ознак і не завжди була достатньою для складних реальних сцен. Особливо це проявлялося в умовах зміни освітлення, різних ракурсів зйомки, шуму та часткового перекриття об'єктів [6].

Сучасні системи комп'ютерного зору дедалі частіше спираються на нейронні мережі, які виконують автоматичне вилучення ознак [7]. Такий підхід дозволяє моделі самостійно формувати багаторівневе представлення вхідного зображення: від простих локальних елементів, таких як краї та контури, до складніших ознак, що відповідають частинам об'єктів або об'єктам загалом. Завдяки цьому нейронна мережа здатна розпізнавати об'єкти в умовах, де класичні алгоритми можуть бути недостатньо надійними (рисунок 1.1).

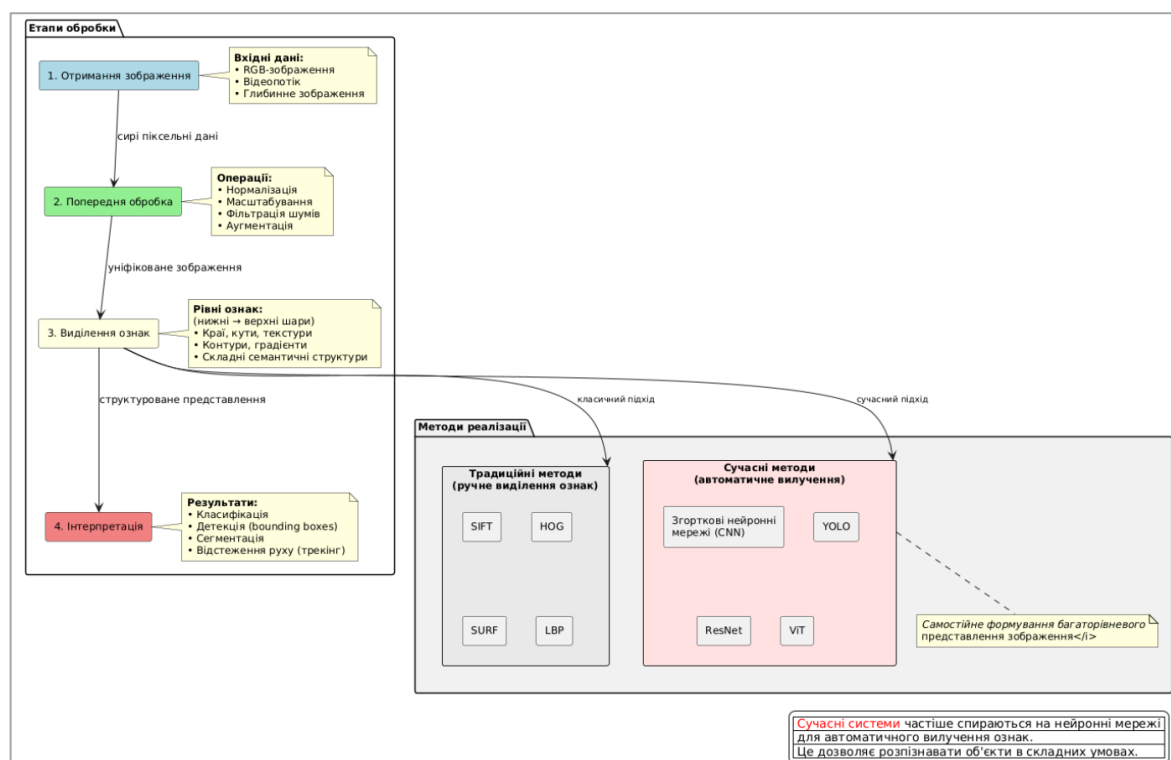


Рисунок 1.1 – Загальна схема роботи системи комп'ютерного зору

У типовій структурі системи комп'ютерного зору можна виділити кілька етапів:

- отримання зображення. Вхідне зображення подається до системи у стандартизованому форматі;
- попередня обробка. Виконується зміна розміру, нормалізація яскравості, вирівнювання контрасту, усунення шумів та інші операції;
- виділення ознак. Модель формує внутрішнє представлення сцени від простих локальних елементів до складніших ознак об'єктів;
- інтерпретація результату. Виконується класифікація зображення, визначення координат об'єктів, сегментація або відстеження руху залежно від задачі.

У контексті задач безпеки найбільш важливими є процеси виявлення об'єктів і аналізу їхнього взаємного розташування. Система має не просто розпізнати наявність предмета, а й визначити, де саме він розташований у межах зображення. Це відкриває можливість для подальшого оцінювання

потенційного ризику на основі геометричних характеристик сцени, зокрема координат обмежувальних рамок, центрів об'єктів і піксельної відстані між ними.

1.3.2 Методи виявлення та класифікації об'єктів

Виявлення об'єктів є однією з центральних задач комп'ютерного зору. Воно передбачає одночасне визначення класу об'єкта та його локалізації на зображенні. На відміну від задачі класифікації, де модель лише встановлює, який об'єкт або клас присутній на зображенні в цілому, детекція об'єктів повертає набір обмежувальних рамок із координатами, що окреслюють кожен знайдений об'єкт. Це особливо важливо для моніторингу безпеки, оскільки потенційний ризик може залежати не лише від наявності небезпечного предмета, а й від його розташування відносно дитини в межах зображення.

У розвитку методів детекції зазвичай виділяють дві великі групи: двоетапні та одноетапні підходи. Двоетапні методи спочатку генерують потенційні області інтересу, а потім уточнюють клас і координати об'єктів у цих областях. Такі підходи можуть демонструвати високу точність, але зазвичай мають більшу обчислювальну складність. Одноетапні методи виконують усі передбачення за один прохід мережі, що забезпечує вищу швидкість і робить їх придатними для задач, де важлива оперативність оброблення зображень [8].

Нейронні мережі стали основою сучасних систем детекції через їхню здатність виявляти складні нелінійні залежності у візуальних даних. Особливо ефективними в задачах аналізу зображень є згорткові нейронні мережі, які дають змогу автоматично добувати просторові ознаки. У процесі навчання мережа поступово формує ієрархію ознак: нижні шари реагують на прості патерни, такі як краї, лінії та текстури, а вищі шари – на складніші структури, що відповідають частинам об'єктів або об'єктам загалом [9].

Для задач безпеки дитини детекція об'єктів має бути не лише точною, а й стабільною в умовах домашнього середовища. Таке середовище часто характеризується змінним освітленням, різноманітним фоном, наявністю дрібних деталей і частковим перекриттям предметів. Наприклад, небезпечний об'єкт може бути частково закритий іграшкою, меблями або рукою дитини. У таких випадках нейронні мережі, особливо навчені на спеціалізованих наборах даних, забезпечують вищу ефективність порівняно з класичними методами, які значною мірою залежать від ручного вибору ознак [10].

Класифікація та детекція об'єктів можуть розглядатися як пов'язані, але різні задачі. Класифікація відповідає на питання, який об'єкт або клас присутній на зображенні, тоді як детекція додатково визначає, де саме цей об'єкт розташований. Для розробленої системи важливою є саме детекція, оскільки вона надає не лише назву класу, а й координати обмежувальної рамки. Надалі ці координати використовуються для визначення центра об'єкта, обчислення піксельної відстані між дитиною та небезпечним предметом і формування висновку про рівень ризику.

Таким чином, методи виявлення об'єктів є основою для побудови інтелектуальної системи моніторингу безпеки дитини. Вони дозволяють перейти від простого розпізнавання зображення до аналізу структури сцени, де враховується не лише наявність об'єктів, а й їхнє взаємне розташування [11].

1.3.3 Застосування нейронних мереж у задачах безпеки

Нейронні мережі відіграють ключову роль у сучасних інтелектуальних системах безпеки, оскільки дають змогу автоматизувати аналіз складних візуальних сцен. Їхня основна перевага полягає в здатності навчатися на великій кількості прикладів і надалі узагальнювати отримані ознаки для нових умов. На відміну від жорстко заданих правил, нейронні

мережі можуть адаптуватися до змін у середовищі, розпізнавати об'єкти різного масштабу та працювати зі складними візуальними ситуаціями.

У задачах безпеки нейронні мережі застосовуються для виявлення небезпечних предметів, аналізу поведінки людей, контролю зон доступу, оцінювання порушень правил перебування в небезпечних ділянках та формування тривожних повідомлень. У випадку дитячого ігрового середовища це означає, що модель може визначати присутність дитини, знаходити у кадрі потенційно небезпечні об'єкти та надавати дані для подальшого аналізу їхнього розташування відносно дитини. Таким чином, нейронна мережа може бути основою для побудови інтелектуального модуля раннього попередження, однак остаточний висновок про ризик формується вже на рівні всієї системи.

Важливою властивістю сучасних нейромережевих систем є можливість швидкого оброблення зображень або окремих кадрів. Для задач безпеки це має практичне значення, оскільки потенційно ризикова ситуація може виникнути раптово, а затримка в аналізі знижує ефективність попередження. Саме тому в прикладних системах безпеки часто обирають архітектури, які поєднують достатню точність із високою швидкістю оброблення візуальних даних.

Ще однією перевагою нейронних мереж є можливість навчання або адаптації на спеціалізованих наборах даних. Для задачі моніторингу безпеки дитини це означає, що модель можна адаптувати до конкретного середовища: домашньої кімнати, дитячої зони або іншого приміщення. При цьому враховуються характерні для такого середовища класи об'єктів. Спеціалізація моделі дозволяє підвищити точність виявлення цільових класів і зменшити кількість хибних спрацювань.

Також у системах безпеки важливу роль відіграє поєднання нейронних мереж із логікою прийняття рішень. Тобто модель не лише виявляє об'єкти, а й передає результати до наступного модуля, який аналізує ризик і визначає, чи потрібно сформулювати попередження. Такий підхід

дозволяє будувати багаторівневі інтелектуальні системи, де нейронна мережа є одним із компонентів загальної архітектури, а не єдиним джерелом остаточного рішення.

У межах даної роботи нейронна мережа використовується для детекції дитини та потенційно небезпечних предметів на зображенні. Вона визначає клас об'єкта, координати обмежувальної рамки та рівень впевненості передбачення. Надалі ці результати передаються до алгоритму оцінювання ризику, який аналізує взаємне розташування об'єктів у межах зображення та формує висновок про потенційно аномальний стан.

1.3.4 Архітектура YOLO та її місце в детекції об'єктів

Серед сучасних нейронних архітектур для детекції об'єктів особливе місце займає сімейство моделей YOLO. Назва You Only Look Once відображає основну ідею підходу: модель аналізує зображення за один прохід нейронної мережі та одразу прогнозує координати обмежувальних рамок, класи об'єктів і рівень впевненості передбачення. Тобто результатом роботи моделі є не просто відповідь про наявність певного об'єкта на зображенні, а повніша інформація про те, який саме об'єкт знайдено, де він розташований і з якою впевненістю модель зробила це передбачення [12].

Такий підхід є важливим для задач детекції, оскільки в багатьох прикладних системах недостатньо лише класифікувати зображення загалом. Наприклад, для системи моніторингу безпеки дитини потрібно не просто визначити, що в кадрі присутній ніж, ліки, ножиці або розетка, а й отримати координати цього об'єкта. Саме координати обмежувальних рамок надалі можуть використовуватися для визначення центра об'єкта, аналізу його положення відносно дитини та обчислення піксельної відстані між ними.

YOLO відрізняється від двоетапних методів детекції тим, що не розділяє процес пошуку об'єктів на окремі великі етапи. У двоетапних підходах спочатку формуються потенційні області інтересу, у яких може

міститися об'єкт, а вже потім виконується уточнення класу та координат. Такі методи можуть забезпечувати високу точність, однак зазвичай потребують більше обчислювальних ресурсів і часу. Натомість YOLO виконує передбачення за один прохід мережі, що дозволяє скоротити час оброблення зображення.

Завдяки такій організації YOLO забезпечує високу швидкість оброблення зображень, що є важливою перевагою для прикладних задач детекції. Для задачі безпеки дитини це має практичне значення, оскільки система повинна швидко отримувати інформацію про наявність дитини та потенційно небезпечних предметів у межах сцени. Навіть у межах прототипу, який працює зі статичними зображеннями, швидкість інференсу є важливою характеристикою, оскільки вона визначає зручність використання системи та можливість її подальшого розширення до аналізу окремих кадрів або відеопотоку.

Архітектура моделей сімейства YOLO зазвичай складається з трьох основних частин: Backbone, Neck та Head. Кожна з цих частин виконує окрему функцію в процесі перетворення вхідного зображення на результати детекції.

Backbone відповідає за початкове вилучення ознак із вхідного зображення. На цьому етапі модель поступово зменшує просторовий розмір карт ознак, але водночас збільшує кількість каналів. Це дозволяє сформувати більш інформативне представлення сцени. На нижчих рівнях мережа виділяє прості візуальні елементи, наприклад краї, контури та текстури, а на глибших рівнях – складніші ознаки, характерні для частин об'єктів або об'єктів загалом.

Neck виконує об'єднання ознак, отриманих на різних рівнях мережі. Це необхідно для того, щоб модель могла ефективно працювати з об'єктами різного розміру. Для задачі безпеки дитини це має особливе значення, оскільки в одному зображенні можуть одночасно бути великі об'єкти, наприклад дитина, і дрібні небезпечні предмети, такі як ліки, ножиці, ніж

або розетка. Об'єднання різномасштабних ознак підвищує здатність моделі працювати зі складними побутовими сценами.

Head є завершальною частиною архітектури та формує кінцеві результати детекції. На цьому етапі модель визначає координати обмежувальних рамок, класи знайдених об'єктів і рівень впевненості для кожного передбачення. Саме ці результати надалі можуть бути використані не лише для візуального відображення знайдених об'єктів, а й для подальшого аналізу їхнього розташування відносно дитини в межах зображення (рисунок 1.2).

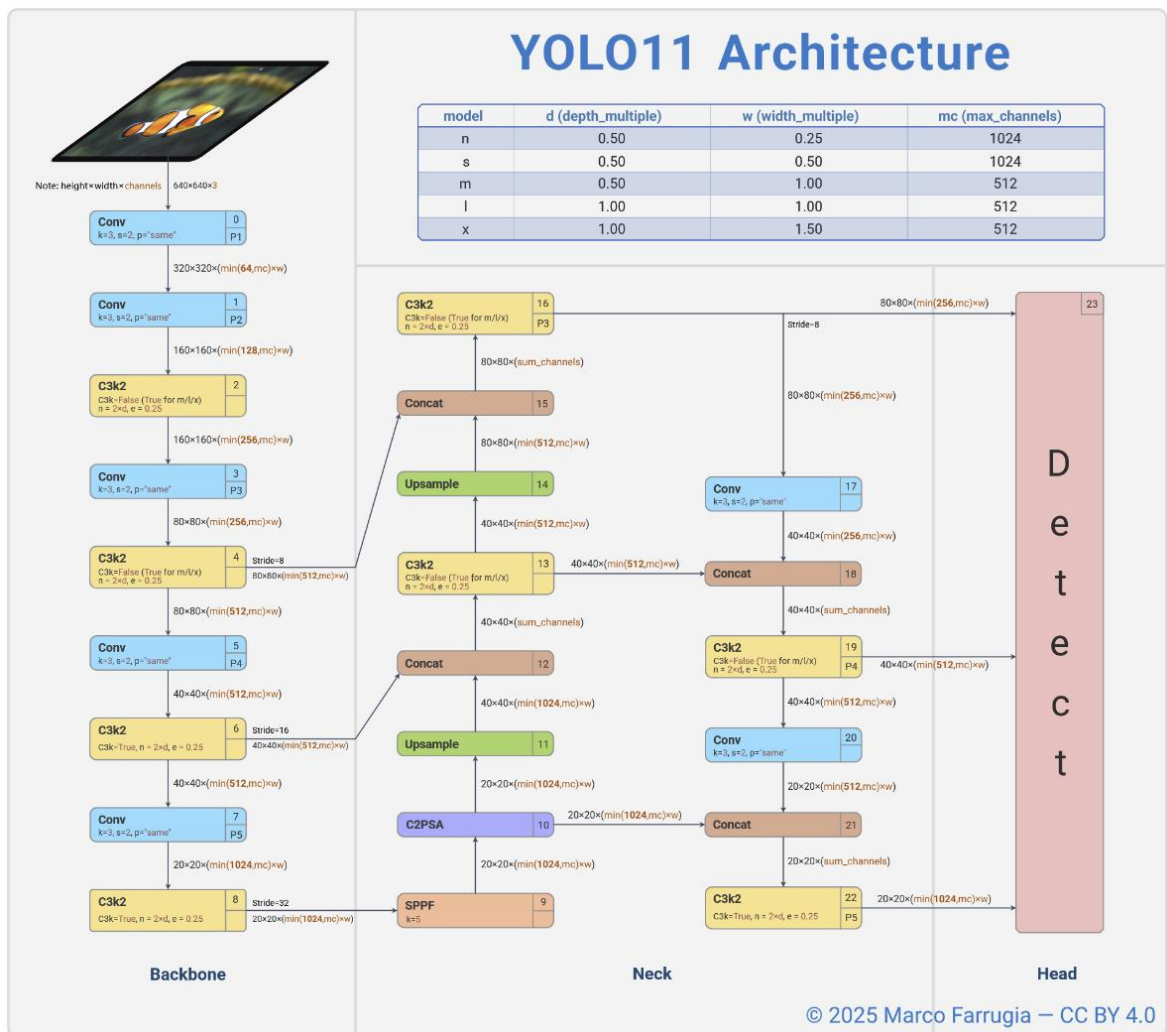


Рисунок 1.2 – Узагальнена архітектура моделі YOLO11 для задачі детекції об'єктів

Саме завдяки такій структурі YOLO добре підходить для задач, де потрібно швидко виконувати детекцію об'єктів на зображеннях або окремих кадрах. У дитячому середовищі це дає змогу розпізнавати присутність дитини й потенційно небезпечних предметів, а також передавати координати виявлених об'єктів до подальшого алгоритму оцінювання ризику. При цьому особливу цінність має здатність моделі працювати з об'єктами різного масштабу, адже небезпечні предмети можуть бути як великими та легко помітними, так і дрібними, наприклад ліками, ножицями або елементами розетки.

У різних версіях YOLO поступово вдосконалювалися точність, швидкодія, стабільність і здатність до роботи з дрібними об'єктами. Нові модифікації моделі спрямовані на покращення балансу між точністю та швидкістю інференсу, підвищення якості локалізації об'єктів і зменшення кількості пропусків. Це особливо важливо для побутових систем безпеки, де невиявлення небезпечного предмета або неточне визначення його рамки може вплинути на подальше оцінювання ризику [13].

Таким чином, архітектура YOLO є доцільною для розроблення системи детекції дитини та небезпечних предметів, оскільки вона поєднує швидке оброблення зображень, визначення класів об'єктів і побудову обмежувальних рамок.

1.3.5 Особливості та переваги YOLO11

YOLO11 є сучасною версією сімейства моделей YOLO, призначеною для розв'язання задач детекції об'єктів, класифікації, сегментації та інших задач комп'ютерного зору. У порівнянні з попередніми версіями, YOLO11 орієнтована на покращення балансу між точністю детекції, швидкодією та обчислювальною ефективністю. Це є важливим для прикладних систем комп'ютерного зору, зокрема для задач моніторингу безпеки, де потрібно

оперативно обробляти зображення та виявляти потенційно небезпечні об'єкти [14].

Для кваліфікаційної роботи було обрано модель YOLO11s, оскільки вона є компактнішою порівняно з більшими варіантами моделі та придатною для експериментального навчання на власному наборі даних. Такий варіант дозволяє зберегти прийнятну якість детекції без надмірного збільшення обчислювальних витрат. Це важливо для прототипу інтелектуальної системи, оскільки модель повинна бути достатньо точною, але водночас зручною для інтеграції у програмний застосунок.

Перевагою YOLO11 для задачі, що розглядається в роботі, є можливість одночасного виявлення кількох об'єктів у межах одного зображення. Це дозволяє системі знаходити не лише дитину, а й кілька потенційно небезпечних предметів у тій самій сцені. Такий підхід є важливим для виявлення потенційно аномальних станів ігрового середовища, оскільки ризик часто залежить не лише від самого факту наявності предмета, а й від його розташування відносно дитини [15].

Також ще однією важливою перевагою є можливість адаптації моделі до спеціалізованого набору даних. У межах цієї роботи модель навчалася на підготовленому датасеті, який містив класи, релевантні для задачі безпеки дитини: baby, knife, medicine, scissors та socket. Завдяки цьому модель була орієнтована саме на виявлення тих об'єктів, які можуть становити потенційний ризик у побутовому ігровому середовищі [16].

Отже, YOLO11s є придатною основою для розроблення інтелектуальної системи моніторингу безпеки дитини. Вона дозволяє виявляти дитину та потенційно небезпечні предмети в межах одного зображення, а отримані результати можуть бути використані для подальшого аналізу взаємного розташування об'єктів і формування висновку про потенційно ризикову ситуацію.

1.4 Методи оцінювання аномальних станів

Після виявлення об'єктів на зображенні наступним етапом інтелектуального аналізу є оцінювання можливих аномальних або потенційно небезпечних станів. У задачах моніторингу безпеки дитини аномальним станом можна вважати не лише появу небезпечного об'єкта в кадрі, а й ситуацію, коли такий об'єкт розташований близько до дитини в межах зображення або потенційно може бути доступним для взаємодії. У цьому випадку ризик визначається не тільки властивостями самого предмета, а й його положенням відносно дитини.

У сучасних системах комп'ютерного зору оцінювання небезпеки часто реалізується через аналіз взаємного розташування розпізнаних об'єктів. Такий підхід дозволяє перетворити результати детекції на логічно інтерпретовані ознаки ризику. Наприклад, якщо модель виявила дитину і небезпечний предмет, система може обчислити піксельну відстань між центрами їхніх обмежувальних рамок, перевірити можливий перетин рамок і на основі заданих правил сформулювати висновок про рівень ризику.

Подібний підхід є доцільним для прототипу системи моніторингу безпеки дитини, оскільки він не потребує складного вимірювання реальної фізичної відстані між об'єктами. Водночас важливо враховувати, що відстань у пікселях не є повним еквівалентом відстані в реальному просторі, адже вона залежить від масштабу зображення, ракурсу камери та перспективи. У зв'язку з цим відносна близькість об'єктів на зображенні розглядається як попередня ознака потенційного ризику, а не як точний показник фізичної відстані [17].

1.4.1 Оцінювання просторової близькості об'єктів

Просторова близькість об'єктів є одним із простих і водночас практично корисних критеріїв для попереднього оцінювання ризику. У

задачі моніторингу безпеки дитини важливо не лише виявити потенційно небезпечний предмет, а й оцінити його розташування відносно дитини на зображенні. Для цього результати детекції, зокрема координати обмежувальних рамок, можуть бути використані для подальшого геометричного аналізу.

У системах комп'ютерного зору взаємне розташування об'єктів може визначатися на основі координат центрів або меж обмежувальних рамок, отриманих у процесі детекції. Якщо на зображенні присутні два об'єкти, наприклад дитина та потенційно небезпечний предмет, то їхнє положення можна подати у вигляді координат у двовимірному просторі кадру. Це дозволяє перейти від простого факту виявлення об'єкта до кількісного аналізу його розташування.

У межах цієї роботи просторову близькість доцільно оцінювати за допомогою евклідової відстані між центрами обмежувальних рамок. Такий підхід є зручним, оскільки модель детекції вже повертає координати знайдених об'єктів. На основі цих координат визначаються центри рамок дитини та потенційно небезпечного предмета, після чого між ними обчислюється піксельна відстань. У двовимірній площині зображення евклідова відстань обчислюється за формулою (1.1):

$$d = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2}, \quad (1.1)$$

де x_1, y_1 – координати центра першого об'єкта;

x_2, y_2 – координати центра другого об'єкта.

У контексті задачі безпеки першим об'єктом зазвичай є дитина, а другим – потенційно небезпечний предмет.

Практичне використання евклідової відстані дозволяє перетворити результати роботи нейронної мережі на числовий показник, який можна використати для подальшого визначення рівня ризику. Якщо піксельна відстань між об'єктами перевищує заданий поріг, система може визначити

низький рівень ризику для поточного зображення. Якщо ж об'єкти розташовані близько один до одного або їхні обмежувальні рамки перетинаються, формується висновок про середній або високий рівень ризику.

Для задач безпеки дитини така близькість є не абсолютною, а відносною характеристикою. Мала піксельна відстань між об'єктами не гарантує фізичного контакту або реальної небезпеки, оскільки результат залежить від масштабу зображення, положення камери, перспективи та розміру об'єктів у кадрі. Водночас вона може слугувати попередньою ознакою потенційно ризикової ситуації. Наприклад, якщо дитина на зображенні розташована поруч із ліками, ножицями, ножом або електричною розеткою, система може позначити таку сцену як таку, що потребує уваги користувача.

Саме тому аналіз близькості доцільно поєднувати з результатами класифікації об'єктів. Наявність у кадрі потенційно небезпечного предмета сама по собі ще не завжди означає ризик, однак у поєднанні з його розташуванням відносно дитини вона може бути підставою для формування попередження. Такий підхід дозволяє враховувати не лише факт виявлення об'єкта, а й просторові характеристики сцени.

Однак використання евклідової відстані має певні обмеження. У розробленому прототипі вона обчислюється у координатах зображення, тобто в пікселях. Такий підхід дозволяє швидко оцінити відносну близькість дитини та небезпечного предмета, але не визначає реальну фізичну відстань між ними у сантиметрах або метрах. Для переходу до вимірювання відстані в реальному просторі необхідно враховувати параметри камери, її положення, масштаб сцени або використовувати додаткові засоби, наприклад камери глибини чи калібрування за еталонними об'єктами [18].

У подальшому цей підхід може бути розширений для роботи з послідовністю кадрів. У такому випадку система могла б аналізувати не лише поточне взаємне розташування об'єктів, а й зміну піксельної відстані

між ними в часі. Якщо відстань між дитиною та небезпечним предметом у кадрі поступово зменшується протягом кількох послідовних кадрів, це могло б бути додатковим індикатором потенційно ризикової ситуації.

Отже, поєднання моделі детекції об'єктів з обчисленням евклідової відстані дозволяє не лише виявити дитину та небезпечний предмет, а й оцінити їхнє взаємне розташування на зображенні. Це забезпечує просту та інтерпретовану логіку попереднього визначення рівня ризику, яка може бути використана для формування попередження про потенційно небезпечну ситуацію.

1.4.2 Формування попереджень про небезпечні ситуації

Формування попереджень є завершальним етапом у ланцюгу аналізу небезпечних ситуацій. Після виявлення об'єктів і обчислення відстані між ними система приймає рішення щодо генерації сигналу тривоги або повідомлення користувачу. У найпростішому випадку це може бути звукове чи візуальне повідомлення про те, що дитина наблизилася до небезпечного предмета. У більш складних реалізаціях попередження може надсилатися на мобільний пристрій, у систему «розумного дому» або до централізованої панелі моніторингу.

Ефективна система попередження повинна враховувати не лише факт наближення, а й частоту спрацювань, щоб уникати надлишкових хибних тривог. Надмірна кількість попереджень знижує довіру користувача до системи та може призвести до ігнорування реальних небезпечних ситуацій. Саме тому важливим є налаштування порогових значень, фільтрація короткочасних спрацювань та, за потреби, використання часової агрегації подій.

У практичних системах безпеки логіка сповіщення часто реалізується за принципом «якщо – то». Якщо відстань між дитиною та небезпечним об'єктом менша за допустимий поріг протягом певного часу, генерується

попередження. Якщо відстань збільшується або небезпечний об'єкт зникає з кадру, тривога може бути скасована. Такий підхід дозволяє зменшити кількість хибних спрацювань, зокрема тих, що виникають через випадкове проходження об'єктів у кадрі або короткочасне перекриття.

Окрему увагу слід приділяти інтерпретованості попереджень. Для користувача важливо не лише отримати сигнал, а й розуміти, чому саме система спрацювала. Тому сучасні інтелектуальні системи дедалі частіше надають не просто повідомлення «небезпека», а пояснення: який об'єкт виявлено, яка відстань між ним і дитиною та який саме критерій був перевищений. Це підвищує довіру до системи і дозволяє користувачу швидше приймати рішення.

Таким чином, формування попередження у системі безпеки дитини ґрунтується на аналізі просторової близькості між об'єктами, перевірці порогових умов і застосуванні логіки фільтрації хибних спрацювань. Це перетворює результати детекції на практичний механізм своєчасного інформування користувача про небезпечну ситуацію.

2 ПОСТАНОВКА ЗАДАЧІ ТА ПРОЄКТУВАННЯ СИСТЕМИ

2.1 Формування вимог до інтелектуальної системи

Після аналізу предметної області та методів оцінювання аномальних станів необхідно сформулювати вимоги до інтелектуальної системи, що розробляється в межах даної роботи. На цьому етапі визначаються основні функції системи, вхідні та вихідні дані, принципи оброблення результатів детекції, вимоги до інтерфейсу користувача, а також обмеження запропонованого підходу.

Розроблювана інтелектуальна система призначена для автоматизованого аналізу зображень ігрового середовища дитини. Її основним завданням є виявлення дитини та потенційно небезпечних об'єктів, визначення їхнього взаємного розташування й формування висновку про наявність або відсутність аномального стану. На відміну від звичайної системи детекції об'єктів, запропонована система не обмежується лише розпізнаванням об'єктів на зображенні, а виконує додаткове оцінювання рівня небезпеки на основі просторового розташування об'єктів.

Вхідними даними системи є зображення побутового або ігрового середовища. У межах кваліфікаційної роботи система орієнтована на оброблення статичних зображень, завантажених користувачем або використаних як тестові приклади. Такий підхід дає змогу перевірити працездатність навченої моделі детекції та алгоритму оцінювання ризику. У подальшому запропонований підхід може бути розширений для аналізу відеопотоку в режимі реального часу.

Основною функціональною вимогою є використання навченої моделі детекції об'єктів. Модель повинна визначати клас кожного виявленого об'єкта, координати його обмежувальної рамки та рівень впевненості передбачення. У межах роботи передбачено розпізнавання таких класів: дитина, ніж, ножиці, медичні препарати та розетка. Дитина розглядається як

основний об'єкт аналізу, а інші зазначені класи – як потенційно небезпечні предмети побутового середовища.

Після отримання результатів детекції система розділяє знайдені об'єкти на дві групи: дитину та небезпечні предмети. Подальший аналіз виконується для кожної пари об'єктів типу «дитина – небезпечний предмет». Якщо на зображенні не виявлено дитину, система не може повноцінно оцінити ризик для неї. Якщо дитина виявлена, але небезпечні об'єкти відсутні, система не формує попередження та повідомляє про відсутність виявлених ризикових предметів.

Важливою вимогою є визначення просторової близькості між об'єктами. Для цього система обчислює центр обмежувальної рамки дитини та центр рамки кожного небезпечного предмета. На основі отриманих координат розраховується відстань у пікселях, яка використовується як кількісний показник близькості об'єктів у межах зображення.

Додатково враховується факт перетину обмежувальних рамок. Якщо рамка дитини перетинається з рамкою небезпечного об'єкта, така ситуація розглядається як така, що має підвищений рівень небезпеки. Це дозволяє враховувати випадки, коли об'єкти перебувають у безпосередній близькості або частково перекривають один одного на зображенні.

У системі передбачено три рівні небезпеки: низький, середній і високий. Низький рівень відповідає ситуації, коли небезпечний об'єкт розташований на відносно безпечній відстані від дитини. Середній рівень визначається у разі наближення небезпечного об'єкта до дитини. Високий рівень встановлюється за умови критичної близькості об'єктів або перетину їхніх обмежувальних рамок.

За результатами аналізу формується загальний висновок про стан середовища. Якщо хоча б для однієї пари «дитина – небезпечний предмет» визначено середній або високий рівень ризику, ситуація класифікується як аномальна. У такому випадку користувачу виводиться попередження про наявність потенційно небезпечної ситуації. Якщо всі виявлені небезпечні

об'єкти розташовані на безпечній відстані від дитини, система повідомляє, що середовище є безпечним у межах виконаного аналізу.

Окремою вимогою є зрозуміле подання результатів роботи системи. Інтерфейс повинен відображати зображення з нанесеними обмежувальними рамками, підписами класів, значеннями впевненості, центрами об'єктів і лініями, що показують відстань між дитиною та небезпечними предметами. Додатково результати доцільно подавати у табличному вигляді, де зазначаються назва небезпечного об'єкта, відстань до дитини, факт перетину рамок і визначений рівень ризику.

До нефункціональних вимог належать точність, швидкодія та зручність використання системи. Точність є важливою, оскільки помилки детекції можуть безпосередньо впливати на правильність оцінювання небезпеки. Швидкодія необхідна для оперативного формування результатів аналізу, а зручність використання передбачає наявність простого інтерфейсу для завантаження зображення, налаштування порогу впевненості та перегляду отриманих результатів.

Під час формування вимог було враховано, що система має працювати з візуальними даними, які можуть істотно відрізнятися за якістю, освітленням, масштабом і кількістю об'єктів. У реальних побутових умовах зображення рідко є ідеальними: дитина може бути частково перекрита меблями або іграшками, небезпечний предмет може займати невелику частину кадру, а фон може містити велику кількість дрібних деталей. Тому система повинна забезпечувати стабільну роботу навіть у складних сценах із частковим перекриттям об'єктів, різним освітленням і деталізованим фоном.

Важливою характеристикою системи є інтерпретованість результатів. Для задачі безпеки недостатньо отримати лише відповідь про наявність або відсутність небезпеки. Користувач повинен розуміти, який саме об'єкт створює ризик, на якій відстані він розташований від дитини та на основі яких ознак система класифікувала ситуацію як безпечну або аномальну.

Саме тому результати аналізу мають супроводжуватися не лише загальним повідомленням, а й візуальними та табличними поясненнями.

Також необхідно передбачити можливість подальшого розширення системи. У цій роботі розглядається обмежений набір класів небезпечних об'єктів, однак архітектура системи не прив'язана лише до цих категорій. За потреби до моделі можна додати нові класи предметів, змінити порогові значення ризику або адаптувати алгоритм до аналізу відеопотоку. Це робить запропонований підхід гнучким для подальших досліджень і практичного вдосконалення.

Водночас необхідно враховувати обмеження розроблюваної системи. Оцінювання відстані виконується в пікселях, а не в реальних одиницях вимірювання, тому результат залежить від масштабу сцени, ракурсу камери та положення об'єктів у кадрі. Для попереднього оцінювання ризику такий підхід є достатнім, однак у подальшому система може бути вдосконалена шляхом калібрування камери, нормалізації відстаней або використання додаткових сенсорів.

Таким чином, сформовані вимоги визначають інтелектуальну систему як програмний комплекс, що поєднує модель детекції об'єктів, алгоритм оцінювання просторової близькості та інтерфейс для відображення результатів. Вихідними даними системи є зображення з результатами детекції, таблиця аналізу ризику та повідомлення про безпечний або аномальний стан ігрового середовища.

2.2 Вибір підходу до реалізації системи та очікувані результати

Для реалізації інтелектуальної системи було обрано підхід, який поєднує нейромережеву детекцію об'єктів і алгоритмічне оцінювання ризику на основі геометричних характеристик сцени. Такий підхід відповідає поставленій задачі, оскільки сама наявність небезпечного предмета в кадрі не завжди означає аномальний стан. Небезпека виникає

тоді, коли об'єкт перебуває поруч із дитиною або потрапляє у зону можливої взаємодії.

Як модель детекції об'єктів обрано YOLO11s. Ця модифікація належить до сімейства YOLO і призначена для швидкого виявлення об'єктів на зображеннях. Офіційна документація Ultralytics розглядає YOLO11 як сучасну модель для задач комп'ютерного зору, зокрема детекції, сегментації, класифікації та інших прикладних задач. Для кваліфікаційної роботи важливими є два фактори: здатність моделі одночасно знаходити кілька об'єктів у кадрі та практична придатність для роботи у режимі, близькому до реального часу.

Вибір саме модифікації YOLO11s обумовлений необхідністю досягти балансу між точністю та обчислювальною складністю. Більші моделі можуть забезпечувати кращу якість розпізнавання, однак потребують більше ресурсів і часу на інференс. Натомість версія s є компромісною для навчального та прикладного прототипу: вона достатньо компактна, але водночас здатна виявляти об'єкти різних класів у складному побутовому середовищі.

Програмну реалізацію виконано мовою Python. Такий вибір є доцільним через наявність розвиненої екосистеми бібліотек для машинного навчання, оброблення зображень і побудови вебінтерфейсів. Для роботи з моделлю YOLO використовується бібліотека Ultralytics, для візуалізації результатів на зображенні – OpenCV, для обчислень – стандартний модуль math, а для створення користувацького інтерфейсу – Streamlit.

Особливістю запропонованого підходу є те, що система не покладає всю логіку безпеки лише на нейронну мережу. Модель виконує завдання локалізації та класифікації об'єктів, а рішення про небезпечність формується окремим алгоритмом. Завдяки цьому висновок системи залишається інтерпретованим: користувач може побачити, який об'єкт був виявлений, яка відстань між ним і дитиною, чи перетинаються рамки та чому сформовано попередження.

Очікуваним результатом реалізації є програмний комплекс, який дозволяє завантажити зображення, виконати детекцію дитини та небезпечних предметів, розрахувати просторові відношення між ними, визначити рівень ризику та вивести результат у зрозумілому вигляді. Результатом роботи системи є не лише зображення з рамками, а й структурований висновок про стан середовища.

Для узагальнення вибраного підходу до реалізації системи основні компоненти програмного комплексу, використані засоби та причини їх вибору наведено в таблиці 2.1.

Таблиця 2.1 – Обґрунтування вибраного підходу до реалізації системи

Компонент	Обраний засіб	Причина вибору
Модель детекції	YOLO11s	Баланс між точністю, швидкістю та можливістю навчання на власному наборі даних
Мова програмування	Python	Підтримка бібліотек для ШІ, комп'ютерного зору та вебпрототипування
Оброблення зображень	OpenCV	Нанесення рамок, точок центрів, ліній відстані та текстових підписів
Інтерфейс	Streamlit	Швидке створення вебзастосунку для завантаження зображень і перегляду результатів
Оцінювання ризику	Пороговий алгоритм	Інтерпретоване рішення на основі відстані та перетину рамок

Таким чином, обраний підхід дозволяє поєднати переваги нейронної детекції об'єктів із простою та зрозумілою логікою оцінювання небезпеки. Модель відповідає за виявлення дитини та потенційно небезпечних предметів, а додатковий алгоритм аналізує їхнє взаємне розташування. Це важливо для задачі превентивного забезпечення безпеки дитини, оскільки

користувач отримує не тільки результат розпізнавання, а й пояснюваний висновок про рівень ризику та причину його формування.

2.3 Загальна архітектура інтелектуальної системи

Загальна архітектура інтелектуальної системи побудована за модульним принципом. Такий підхід дозволяє розділити процес оброблення зображення на окремі логічні етапи: отримання вхідних даних, детекцію об'єктів, підготовку результатів моделі, оцінювання ризику, визначення аномального стану та відображення результату користувачу.

Першим компонентом архітектури є модуль введення даних. У вебверсії системи цей модуль реалізовано через механізм завантаження зображення користувачем. Користувач може обрати файл у форматі jpg, jpeg або png, після чого зображення перетворюється у формат RGB та передається до моделі детекції.

Другим компонентом є модуль детекції об'єктів. Він завантажує навчену модель YOLO11s і виконує інференс на вхідному зображенні. На виході моделі формується набір детекцій, кожна з яких містить номер класу, назву класу, координати обмежувальної рамки та рівень впевненості. Саме ці значення є базою для подальшої логіки оцінювання небезпеки.

Третім компонентом є модуль структуризації детекцій. Він виконує розділення об'єктів на дві групи: об'єкти класу baby та об'єкти небезпечних класів. Таке рішення робить систему простою для подальшого розширення: у разі додавання нового небезпечного класу достатньо оновити відповідний список класів і перенавчити модель на розширеному наборі даних.

Четвертим компонентом є модуль оцінювання ризику. Він перебирає всі пари «дитина – небезпечний предмет», визначає центр кожної рамки, обчислює відстань між центрами, перевіряє перетин рамок і присвоює рівень ризику. Для кожної пари формується окремий запис, який у подальшому використовується для таблиці аналізу.

П'ятим компонентом є модуль визначення аномального стану. Його завдання полягає у перетворенні локальних оцінок ризику на загальний висновок щодо стану середовища. Якщо хоча б одна пара об'єктів має середній або високий рівень ризику, система вважає, що аномальний стан виявлено. Якщо таких пар немає, середовище позначається як безпечне в межах виконаного аналізу.

Шостим компонентом є модуль візуалізації та інтерфейсу. Він відповідає за показ результату користувачу: відображення початкового та обробленого зображення, кількості виявлених дітей і небезпечних об'єктів, кількості оцінених пар ризику, а також таблиці з деталізованим аналізом. Така структура робить результат зрозумілим і придатним для перевірки. Основні модулі архітектури інтелектуальної системи, їхнє призначення та результати роботи наведено в таблиці 2.2.

Таблиця 2.2 – Основні модулі архітектури інтелектуальної системи

Модуль	Призначення	Результат роботи
Введення даних	Завантаження та підготовка зображення	RGB-зображення для моделі
Детекція об'єктів	Виявлення класів і координат рамок	Список об'єктів з class, conf, хуху
Структуризація детекцій	Поділ на дитину та небезпечні об'єкти	Списки babies і danger_objects
Оцінювання ризику	Розрахунок відстані та перетину рамок	Рівень ризику для кожної пари
Виявлення аномалії	Узагальнення результатів оцінювання	Ознака anomaly_detected
Інтерфейс користувача	Відображення результатів	Зображення, метрики, таблиця аналізу

Архітектура системи має послідовний характер, однак окремі модулі є логічно незалежними. Це дозволяє вдосконалювати систему поетапно: наприклад, замінити модель YOLO11s на іншу версію, змінити пороги

ризик, додати нові класи небезпечних предметів або розширити інтерфейс без повної перебудови програмного забезпечення.

Модульний підхід є доцільним ще й тому, що кожен компонент системи виконує чітко визначену функцію. Модуль детекції відповідає за виявлення об'єктів, модуль оцінювання ризику працює з координатами та геометричними характеристиками, а модуль інтерфейсу забезпечує взаємодію з користувачем. Такий поділ дозволяє уникнути змішування логіки машинного навчання, оброблення зображень і представлення результатів.

У разі вдосконалення системи окремі модулі можуть бути замінені без повної зміни архітектури. Наприклад, замість YOLO11s можна використати іншу модель детекції, зберігши той самий алгоритм аналізу просторової близькості. Так само можна змінити вебінтерфейс або додати модуль сповіщень, не змінюючи принципу роботи детекційного ядра.

Важливо, що запропонована архітектура передбачає перехід від низькорівневих результатів нейронної мережі до зрозумілого для користувача висновку. Модель повертає координати, класи та значення впевненості, однак ці дані самі по собі ще не є рішенням про безпеку. Лише після їх структуризації, порівняння об'єктів і розрахунку відстаней формується узагальнений висновок про стан середовища.

2.4 Визначення класів об'єктів для детекції

Визначення класів об'єктів є важливим етапом проектування системи, оскільки саме від набору класів залежить, які ситуації можуть бути виявлені автоматично. У межах роботи обрано п'ять класів: `baby`, `knife`, `medicine`, `scissors` та `socket`. Назви класів залишено англійською, оскільки саме в такому вигляді вони використовуються у структурі набору даних, файлі конфігурації `data.yaml` та результатах передбачення моделі YOLO11s. Це дозволяє уникнути додаткового перетворення міток під час програмної

обробки результатів детекції. Клас `baby` відповідає дитині, а решта класів належать до потенційно небезпечних об'єктів.

Такий набір класів обрано не випадково. Ніж і ножиці є гострими предметами, що можуть призвести до механічного травмування. Медичні препарати пов'язані з ризиком отруєння або неправильного вживання. Розетка є джерелом електричної небезпеки, особливо для дитини раннього віку. Українські та міжнародні рекомендації з безпеки дітей окремо наголошують на необхідності обмеження доступу дітей до ліків, гострих предметів, електричних джерел та інших небезпечних об'єктів.

Безпечні об'єкти, наприклад іграшки, меблі або інші побутові предмети, не виділялися в окремі класи, оскільки вони не використовуються в алгоритмі формування попередження. Метою системи є не повний опис усієї сцени, а виявлення дитини та об'єктів, які можуть створювати потенційну небезпеку. Такий підхід дозволяє зосередити навчання моделі на найбільш важливих для поставленої задачі класах і не ускладнювати систему зайвими категоріями.

У системі клас `baby` є центральним, оскільки саме від його просторового розташування оцінюється ризик. Небезпечний об'єкт не створює аномального стану сам по собі, якщо дитина не перебуває поруч. Тому всі алгоритмічні перевірки виконуються відносно координат рамки дитини.

Класи `knife`, `medicine`, `scissors` і `socket` у програмі згруповані в окремий список небезпечних об'єктів. Таке рішення спрощує логіку оброблення: модель може повертати різні класи, але алгоритм ризику аналізує лише ті з них, які входять до визначеного списку. Це також дозволяє в майбутньому додавати нові класи, наприклад `lighter`, `cable`, `chemical`, `glass` або `battery`, не змінюючи загальну логіку програми.

Таким чином, обрані класи відображають базові побутові ризики, що можуть виникати в ігровому середовищі дитини. Їх достатньо для демонстрації роботи системи та перевірки запропонованого підходу до

виявлення аномальних станів. При цьому обраний набір не претендує на повне охоплення всіх можливих небезпек, а використовується як базовий набір для побудови й перевірки прототипу інтелектуальної системи.

2.5 Розроблення алгоритму оцінювання рівня небезпеки

Алгоритм оцінювання рівня небезпеки є одним із ключових компонентів розроблюваної інтелектуальної системи, оскільки забезпечує перетворення результатів нейромережевої детекції на інтерпретований висновок про потенційний ризик у межах зображення. На вхід алгоритм отримує координати обмежувальних рамок дитини та потенційно небезпечних об'єктів, виявлених моделлю YOLO. На виході формується один із трьох рівнів ризику: низький, середній або високий.

Першим етапом роботи алгоритму є визначення центра кожної обмежувальної рамки. Для цього використовуються координати рамки, які повертає модель детекції. Центр об'єкта визначається як середина відповідної рамки по горизонталі та вертикалі. Такий підхід дозволяє подати кожен виявлений об'єкт у вигляді однієї точки в координатній площині зображення, що спрощує подальший аналіз його розташування.

Другим етапом є обчислення відстані між центром рамки дитини та центром рамки небезпечного об'єкта. У межах програмної реалізації використовується евклідова відстань у пікселях, яка дає змогу кількісно оцінити просторову близькість об'єктів у межах кадру. Отримане значення використовується як один з основних показників для подальшого визначення рівня ризику.

Додатково алгоритм перевіряє факт перетину обмежувальних рамок. Така перевірка необхідна для ситуацій, коли центри рамок можуть бути розташовані на певній відстані, але самі об'єкти вже візуально накладаються один на одного або перебувають у безпосередній близькості. У разі

перетину рамок система встановлює високий рівень ризику незалежно від значення відстані між центрами.

У розробленій системі використано такі правила:

- високий ризик. Відстань між дитиною та небезпечним об'єктом не перевищує 120 px або обмежувальні рамки об'єктів перетинаються;
- середній ризик. Відстань перевищує 120 px, але не перевищує 250 px;
- низький ризик. Відстань більша за 250 px і перетину обмежувальних рамок не виявлено.

Порогові значення 120 px і 250 px були обрані експериментально з урахуванням розміру тестових зображень і зручності демонстрації роботи прототипу. Такий поділ дозволяє умовно розмежувати ситуації, у яких небезпечний об'єкт розташований дуже близько до дитини, перебуває на помірній відстані або не становить безпосереднього ризику в межах аналізованого кадру.

Перевагою запропонованого алгоритму є його простота, швидкодія та інтерпретованість. Користувач або розробник може простежити, на основі яких саме умов система визначила конкретний рівень небезпеки. Це важливо для задач превентивного моніторингу, оскільки результат роботи системи має бути зрозумілим і придатним для подальшого аналізу.

Водночас алгоритм має певні обмеження, оскільки оцінювання виконується у площині зображення, а відстань вимірюється в пікселях. Однакова піксельна відстань може відповідати різним фізичним відстаням залежно від масштабу сцени, ракурсу камери, положення об'єктів і перспективи. У межах кваліфікаційної роботи такий підхід є достатнім для демонстраційного прототипу, однак у подальшому його можна вдосконалити шляхом нормалізації відстані відносно розміру рамки дитини, калібрування камери або використання сенсорів глибини.

Таким чином, розроблений алгоритм оцінювання рівня небезпеки дозволяє на основі координат виявлених об'єктів визначати ступінь

потенційного ризику для кожної пари «дитина – небезпечний об’єкт». Отримані результати надалі використовуються для формування загального висновку про наявність або відсутність аномального стану в ігровому середовищі.

2.6 Розроблення алгоритму виявлення аномальних станів

Алгоритм виявлення аномальних станів призначений для формування загального висновку про стан ігрового середовища на основі результатів детекції та оцінювання ризику. Якщо алгоритм, описаний у підрозділі 2.5, визначає рівень небезпеки для окремої пари об’єктів, то алгоритм виявлення аномального стану узагальнює ці результати для всього зображення.

Після виконання нейромережевої детекції система формує два набори об’єктів: список виявлених дітей і список потенційно небезпечних предметів. До небезпечних об’єктів у межах розробленої системи належать класи *knife*, *medicine*, *scissors* та *socket*. Далі виконується послідовне порівняння кожної виявленої дитини з кожним небезпечним об’єктом. Такий підхід дає змогу аналізувати не лише прості сцени з одним об’єктом, а й складніші випадки, коли в кадрі присутні кілька потенційних джерел ризику.

Для кожної сформованої пари «дитина – небезпечний об’єкт» застосовується алгоритм оцінювання рівня небезпеки. У результаті система отримує значення ризику для кожної пари об’єктів. Якщо хоча б одна пара має середній або високий рівень ризику, змінна *anomaly_detected* набуває значення *True*. Це означає, що в межах поточного зображення виявлено потенційно аномальний стан, який потребує уваги користувача.

Якщо всі сформовані пари мають низький рівень ризику, аномальний стан не фіксується. У випадку, коли в кадрі не виявлено дитину або не знайдено жодного небезпечного об’єкта, система не може сформувати повноцінну пару для оцінювання ризику. Такий результат інтерпретується

окремо: за відсутності небезпечних об'єктів ситуація не вважається ризиковою, а за відсутності дитини аналіз взаємного розташування не виконується.

Важливою особливістю алгоритму є збереження не лише підсумкового результату, а й деталізованої інформації про кожну проаналізовану пару. До такої інформації належать назва небезпечного об'єкта, рівень впевненості детекції, розрахована відстань, наявність або відсутність перетину рамок, а також визначений рівень ризику. Це підвищує прозорість роботи системи, оскільки користувач може зрозуміти, який саме об'єкт вплинув на формування попередження.

У разі виявлення аномального стану система відображає відповідне повідомлення в інтерфейсі користувача. Додатково на результуюче зображення виводиться напис ANOMALY DETECTED, що забезпечує швидке візуальне сприйняття результату аналізу. Така індикація є зручною під час тестування системи, оскільки дозволяє одразу визначити, чи було поточне зображення класифіковане як потенційно небезпечне.

До аномальних станів у розробленій системі віднесено не лише ситуації з високим, а й із середнім рівнем ризику. Це пояснюється превентивним характером системи: її завдання полягає у ранньому виявленні потенційно небезпечного розташування об'єктів, а не лише у фіксації критичної ситуації. Такий підхід дозволяє сформувати попередження до моменту можливого контакту дитини з небезпечним предметом.

Таким чином, алгоритм виявлення аномальних станів виконує узагальнення результатів детекції та оцінювання ризику для всього зображення. Він дозволяє перейти від набору окремих знайдених об'єктів до підсумкового висновку про стан ігрового середовища, що є необхідним для роботи інтелектуальної системи превентивного забезпечення безпеки дитини.

2.7 Проєктування структури користувацького інтерфейсу

Користувацький інтерфейс розроблюваної системи має забезпечити просту взаємодію користувача з моделлю детекції та результатами аналізу. Оскільки система створюється як прототип для демонстрації роботи алгоритму, інтерфейс повинен бути зрозумілим, мінімалістичним і не вимагати від користувача роботи з командним рядком або програмним кодом.

Для побудови інтерфейсу обрано Streamlit. Цей інструмент дозволяє швидко створювати вебзастосунки мовою Python і поєднувати їх із моделями машинного навчання. У системі Streamlit використано для створення сторінки, завантаження зображення, налаштування порогу впевненості, відображення кількісних показників, вкладок із результатами та таблиці аналізу ризику.

Інтерфейс умовно поділено на дві основні області: область налаштувань і область результатів. Область налаштувань призначена для введення даних: користувач завантажує зображення та встановлює поріг впевненості моделі. Також у цій області подано коротку інформацію про класи моделі та пороги ризику. Область результатів призначена для виведення статусу середовища, кількості виявлених об'єктів, результату детекції та таблиці аналізу.

Поріг впевненості моделі реалізовано у вигляді повзунка. Це дозволяє користувачу змінювати чутливість детекції без редагування коду. Нижче значення порогу може збільшувати кількість виявлених об'єктів, але водночас підвищувати ризик хибних спрацювань. Вище значення порогу робить результат суворішим, однак може призвести до пропуску слабо виражених об'єктів.

Виведення результатів організовано за допомогою вкладок. Перша вкладка містить оброблене зображення з рамками, підписами та лініями відстаней. Друга вкладка показує оригінальне зображення, що дозволяє

порівняти вхідні та вихідні дані. Третя вкладка містить таблицю аналізу ризику, де наведено назву об'єкта, впевненість моделі, відстань до дитини, факт перетину рамок і рівень ризику.

Окремо в інтерфейсі виводяться кількісні показники: кількість виявлених дітей, кількість небезпечних об'єктів і кількість оцінених пар ризику. Ці показники не замінюють детальний аналіз, але дають користувачу швидке уявлення про структуру сцени та обсяг виконаної перевірки.

Побудова інтерфейсу за таким принципом також зменшує навантаження на користувача. Усі основні елементи керування розміщені в одному місці, а результати аналізу представлені у вигляді, який не потребує спеціальних знань із машинного навчання. Це важливо, оскільки потенційними користувачами такої системи можуть бути не лише розробники, а й люди, які потребують простого інструменту для перевірки зображень.

Окремою перевагою є можливість швидко змінювати поріг впевненості моделі під час тестування прототипу. Це дозволяє порівнювати результати при різних налаштуваннях і оцінювати, як змінюється кількість виявлених об'єктів. Завдяки цьому інтерфейс виконує не лише демонстраційну, а й експериментальну функцію.

Таким чином, структура інтерфейсу відповідає логіці роботи системи: спочатку користувач завантажує зображення, після чого система виконує детекцію, аналізує ризик і подає результат у візуальній та табличній формах. Такий підхід забезпечує зручність перевірки роботи моделі та дозволяє швидко оцінити, чи присутній у сцені потенційно ризиковий стан у межах виконаного аналізу. Це робить систему придатною для демонстрації, тестування та подальшого розвитку.

3 РОЗРОБЛЕННЯ ТА НАВЧАННЯ МОДЕЛІ ДЕТЕКЦІЇ ОБ'ЄКТІВ

3.1 Формування та підготовка набору даних

Практична частина роботи передбачала формування комбінованого спеціалізованого набору даних для задачі детекції дитини та потенційно небезпечних об'єктів у побутовому або ігровому середовищі. Оскільки наявні відкриті набори даних не повністю відповідають поставленій задачі, підготовка датасету виконувалася з урахуванням предметної області кваліфікаційної роботи, цільових класів системи та необхідності наявності складних побутових сцен. Такий підхід дозволив сформувати набір даних, орієнтований не на загальну детекцію об'єктів, а саме на виявлення ситуацій, пов'язаних із потенційними ризиками для дитини.

Підготовка набору даних виконувалася на платформі Roboflow. На першому етапі було об'єднано кілька наявних наборів даних, що містили зображення з релевантними для роботи об'єктами. Після імпорту виконувалася перевірка анотацій і уніфікація класів: у випадках, коли клас на зображенні був зазначений некоректно або не відповідав прийнятій схемі `baby`, `knife`, `medicine`, `scissors`, `socket`, мітки виправлялися вручну. Додатково було самостійно відібрано, завантажено та розмічено складні сцени, у яких дитина або небезпечні предмети розташовані на складному фоні, частково перекриті іншими об'єктами або мають малий розмір у кадрі. Це було необхідно для того, щоб модель навчалася не лише на простих прикладах, а й на зображеннях, наближених до реальних умов використання системи.

Використання Roboflow було доцільним, оскільки платформа підтримує повний цикл роботи з датасетом: об'єднання та організацію зображень, ручну розмітку об'єктів, редагування класів, перевірку анотацій, створення версій датасету, застосування аугментацій і подальший експорт у форматі, сумісному з моделями YOLO. Для навчання моделі було використано формат YOLO, у якому кожному зображенню відповідає

текстовий файл з анотаціями обмежувальних рамок. Такий формат є зручним для подальшого навчання засобами Ultralytics YOLO, оскільки безпосередньо містить інформацію про клас об'єкта та координати його розташування на зображенні.

У процесі підготовки даних особлива увага приділялася точності меж об'єктів і правильності призначення класів. Для задачі оцінювання небезпеки координати рамок мають значення не лише для навчання моделі, а й для подальшого програмного аналізу: саме з них обчислюються центри об'єктів, відстані між ними та факт перетину рамок. Тому некоректна розмітка або помилково заданий клас могли б вплинути не тільки на якість детекції, але й на правильність визначення рівня ризику.

Оскільки датасет був комбінованим, важливо було привести його до єдиної структури класів. У різних наборах даних однакові об'єкти могли мати різні назви або різний принцип розмітки, тому під час підготовки виконувалося уточнення класів і перевірка відповідності позначень. Це дозволило уникнути ситуацій, коли модель навчається на суперечливих мітках і гірше розпізнає об'єкти одного типу. Єдина структура класів також спростила подальшу інтеграцію навченої моделі в програмний застосунок, де ці класи використовуються для розділення об'єктів на дитину та небезпечні предмети.

Додані складні сцени мали важливе значення для підвищення практичної цінності датасету. У реальному ігровому середовищі небезпечні предмети не завжди розташовані ізольовано або на нейтральному фоні. Вони можуть бути частково закриті іншими предметами, розміщені поруч із іграшками або мати малий розмір у кадрі. Саме такі приклади допомагають моделі краще адаптуватися до умов, наближених до реального використання. Крім того, наявність різноманітних сцен зменшує ризик того, що модель буде добре працювати лише на однотипних зображеннях.

Підготовка датасету також впливає на подальшу роботу модуля оцінювання ризику. Якщо рамка об'єкта побудована неточно, то центр

рамки буде зміщений, а розрахована відстань між дитиною та небезпечним предметом може бути некоректною. Тому якість розмітки є важливою не лише для самої детекції, а й для всього алгоритму визначення аномального стану. Таким чином, етап підготовки даних безпосередньо впливає на точність роботи всієї інтелектуальної системи. Основні етапи підготовки набору даних у середовищі Roboflow наведено в таблиці 3.1.

Таблиця 3.1 – Етапи підготовки набору даних у Roboflow

Етап	Зміст робіт	Результат
Об'єднання наборів даних	Поєднання кількох релевантних датасетів у робочому проєкті Roboflow	Сформовано комбіновану основу для навчального набору
Перевірка та уніфікація класів	Перегляд анотацій і виправлення класів там, де вони не відповідали прийнятій схемі	Узгоджено класи baby, knife, medicine, scissors, socket
Додавання складних сцен	Самостійний пошук, відбір і завантаження додаткових зображень зі складним фоном, перекриттями або малими об'єктами	Підвищено репрезентативність датасету
Ручна розмітка	Виділення дитини та небезпечних об'єктів обмежувальними рамками	Створено або уточнено анотації для релевантних об'єктів
Генерація версії та аугментація	Формування версії датасету в Roboflow із використанням налаштувань підготовки й аугментації	Отримано розширений набір даних для навчання
Експорт	Вивантаження набору даних у форматі YOLO	Отримано структуру, сумісну з Ultralytics YOLO

3.2 Аналіз структури датасету

Після експорту з Roboflow набір даних мав структуру, характерну для навчання моделей YOLO. Він містив окремі каталоги із зображеннями та файлами анотацій, а також конфігураційний файл `data.yaml`. У цьому файлі задаються шляхи до навчальної, валідаційної та тестової вибірок, кількість класів і назви класів, які модель повинна розпізнавати.

У межах розробленої системи було визначено п'ять класів об'єктів: `baby`, `knife`, `medicine`, `scissors` та `socket`. Для кожного зображення з розміткою створювався відповідний файл анотації, у якому зберігалися номер класу та нормалізовані координати обмежувальної рамки. Такий формат є компактним і безпосередньо підтримується середовищем Ultralytics YOLO. Перевагою цього підходу є те, що один і той самий набір даних можна використовувати як для навчання моделі, так і для подальшої перевірки результатів детекції.

Візуальний аналіз файлу `labels.jpg` дозволяє оцінити загальне розміщення та розподіл обмежувальних рамок у наборі даних. Такі діаграми є корисними для перевірки того, чи має датасет помітні перекоси за кількістю класів, положенням або розміром об'єктів.

За результатами аналізу видно, що найбільшу кількість екземплярів має клас `medicine`, тоді як найменшу – клас `scissors`. Клас `baby` також представлений значною кількістю прикладів, що є важливим для задачі, оскільки саме дитина є центральним об'єктом подальшого аналізу ризику. Такий розподіл свідчить про певну нерівномірність класів у наборі даних, однак усі цільові категорії представлені достатньою кількістю прикладів для навчання моделі.

Діаграми розташування обмежувальних рамок показують, що значна частина об'єктів зосереджена ближче до центральної області зображення. Це є типовим для побутових сцен і зображень із цільовими об'єктами, оскільки під час збирання або розмітки даних основний об'єкт часто

розташовується в центрі кадру. Водночас наявність рамок у різних частинах зображення свідчить про те, що модель отримує приклади з різним положенням об'єктів.

Розподіл ширини та висоти рамок демонструє наявність об'єктів різного масштабу. Це важливо для поставленої задачі, оскільки дитина зазвичай займає більшу частину кадру, тоді як небезпечні предмети можуть бути значно меншими за розміром. Наявність таких прикладів дозволяє моделі навчатися виявляти як великі, так і дрібні об'єкти, що підвищує практичну цінність підготовленого набору даних.

Узагальнений візуальний розподіл класів, положення центрів об'єктів і розмірів обмежувальних рамок наведено на рисунку 3.1.

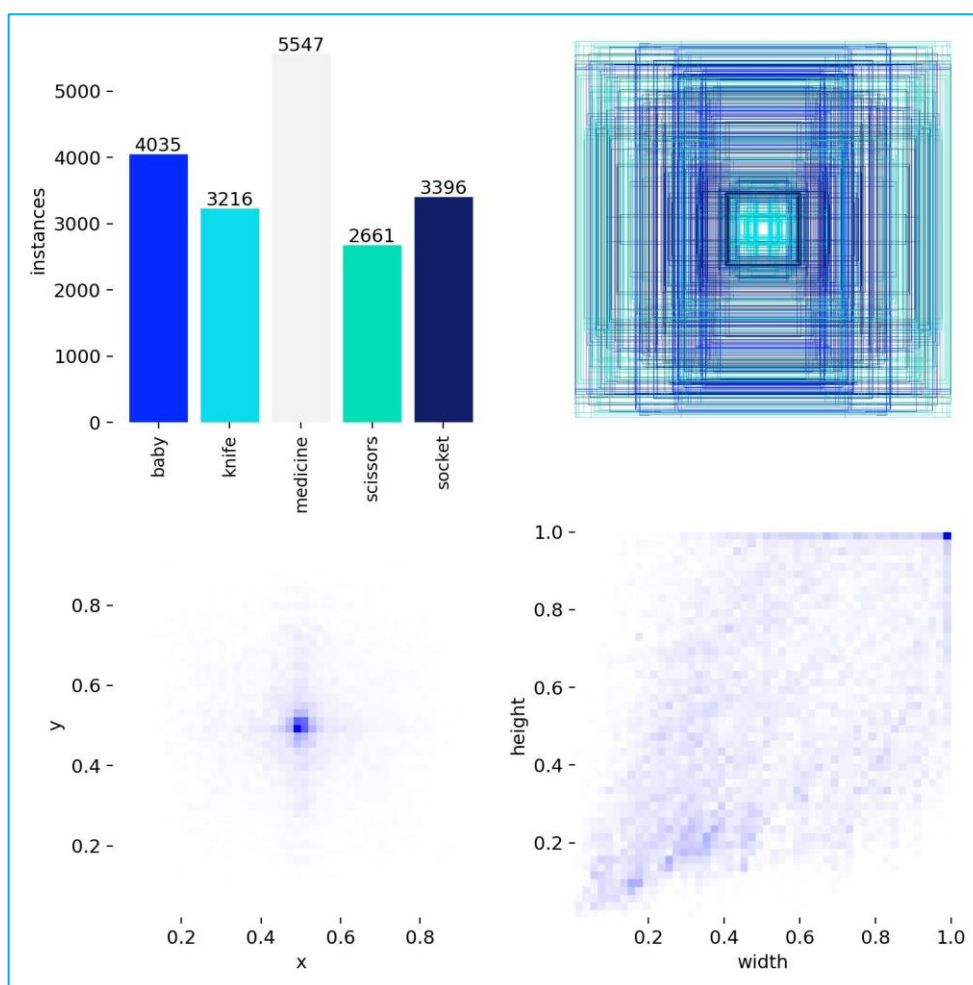


Рисунок 3.1 – Візуальний аналіз розподілу анотацій у наборі даних

3.3 Розподіл даних на навчальну, валідаційну та тестову вибірки

Для коректного навчання та оцінювання моделі датасет було розділено на навчальну, валідаційну та тестову вибірки. Навчальна вибірка використовується для оновлення ваг нейронної мережі, валідаційна – для контролю якості моделі під час навчання, а тестова – для підсумкової перевірки працездатності моделі на прикладах, які не використовувалися для підбору параметрів.

Розподіл даних виконувався на етапі підготовки версії датасету в Roboflow після об'єднання наборів, уточнення класів і перевірки розмітки. Це дозволило автоматично сформувати необхідну структуру каталогів і забезпечити відповідність між зображеннями та файлами анотацій. Важливо, що розділення даних зменшує ризик переоцінювання якості моделі, оскільки модель оцінюється також на відокремлених прикладах, які не використовувалися для оновлення її ваг.

У сформованій версії датасету загальна кількість зображень становила 14807 одиниць. Розподіл даних було виконано на три частини: навчальну, валідаційну та тестову вибірки. Навчальна вибірка містила 13215 зображень, що становить 89% від загального обсягу даних. Валідаційна вибірка містила 1005 зображень, тобто 7%, а тестова вибірка – 587 зображень, що становить 4% від загальної кількості.

Такий розподіл є доцільним для навчання моделі детекції об'єктів, оскільки основна частина даних використовується для формування ознак і оновлення ваг нейронної мережі. Водночас окремі валідаційна та тестова вибірки дозволяють контролювати якість моделі на прикладах, які не використовуються безпосередньо для навчання. Це дає змогу об'єктивніше оцінити здатність моделі працювати з новими зображеннями.

У задачі детекції небезпечних ситуацій важливо, щоб у кожній вибірці були представлені приклади з різними класами, масштабами об'єктів і фоновими умовами. Це підвищує здатність моделі узагальнювати

результати та зменшує ймовірність того, що вона буде працювати коректно лише для обмеженого набору зображень.

Навчальна вибірка повинна містити достатньо прикладів для формування ознак кожного класу. Саме на її основі модель поступово навчається розпізнавати форми, контури, текстури та інші характерні властивості об'єктів. Якщо певний клас представлений недостатньо або занадто одноманітно, модель може гірше узагальнювати результати на нових зображеннях.

Валідаційна вибірка використовується для контролю якості моделі під час навчання. Вона дозволяє оцінити, чи покращується модель не лише на тих зображеннях, за якими оновлюються ваги, а й на відокремлених прикладах. Це важливо для виявлення можливого перенавчання, коли модель добре запам'ятовує навчальні зображення, але гірше працює з новими сценами.

Тестові приклади потрібні для підсумкової перевірки працездатності системи. У межах цієї роботи вони використовувалися для демонстрації якості детекції, аналізу типових помилок і перевірки роботи вебзастосунку. Такий поділ дозволяє відокремити процес навчання від процесу оцінювання результатів і зробити висновки більш обґрунтованими.

3.4 Аугментація даних та її вплив на процес навчання

Аугментація даних застосовувалася для підвищення варіативності навчального набору та зменшення ризику перенавчання. У межах роботи аугментація виконувалася під час формування версії датасету в Roboflow після об'єднання кількох наборів, виправлення класів і додавання складних сцен. Це дало змогу розширити комбінований набір даних і сформувати додаткові навчальні приклади для більш різноманітних умов зйомки без повністю ручного створення кожного нового зображення. Такий підхід є доцільним, оскільки ручне збирання великої кількості побутових сцен із

різними умовами освітлення, фоном і положенням об'єктів потребує значних часових витрат [19].

Для задачі моніторингу безпеки дитини аугментація є особливо важливою, оскільки реальні умови зйомки можуть істотно відрізнятися. У побутовому середовищі змінюється освітлення, фон, масштаб об'єктів, відстань до камери та взаємне розташування предметів. Крім того, небезпечні предмети можуть бути розташовані під різними кутами, частково перекриватися іншими об'єктами або займати невелику частину кадру. Тому модель повинна навчатися не лише на однакових або дуже схожих кадрах, а й на прикладах із різними візуальними характеристиками.

У Roboflow було використано створення трьох вихідних варіантів для одного навчального прикладу. Для підвищення варіативності датасету застосовувалися такі перетворення:

- горизонтальне та вертикальне віддзеркалення;
- поворот зображення в межах від -10° до $+10^\circ$;
- зміна яскравості в межах від -15% до $+15\%$;
- зміна експозиції в межах від -10% до $+10\%$;
- розмиття зображення до 2,5 px.

Такі перетворення дозволили імітувати частину змін, які можуть виникати під час реального отримання зображень: різне освітлення, незначний нахил камери, зміну чіткості та положення об'єктів у кадрі. Завдяки цьому модель отримує більше різноманітних прикладів одного й того самого класу, що сприяє кращому узагальненню під час роботи з новими зображеннями.

Окрім аугментацій, виконаних під час формування версії датасету в Roboflow, під час навчання в Ultralytics YOLO також використовувалися стандартні навчальні перетворення. У файлі параметрів навчання зафіксовано використання таких налаштувань, як `hsv_h`, `hsv_s`, `hsv_v`, `translate`, `scale`, `fliplr`, `mosaic`, `auto_augment` та `erasing`. Ці параметри відповідають за зміну колірних характеристик, масштабування, зсуви,

горизонтальне віддзеркалення, поєднання кількох зображень у межах одного навчального прикладу та часткове приховування ділянок зображення. У таблиці 3.2 подано аугментації, використані під час підготовки даних та навчання моделі, а також їх очікуваний вплив на результат.

Таблиця 3.2 – Аугментації та їх очікуваний вплив на модель

Тип перетворення	Призначення	Очікуваний ефект
Зміна освітлення та кольору	Імітація різних умов зйомки	Стійкість до освітлення та тіней
Масштабування	Зміна видимого розміру об'єктів	Краще виявлення дрібних і великих об'єктів
Зсуви та зміни положення	Імітація різного розміщення об'єктів у кадрі	Менша залежність від позиції предмета
Горизонтальне віддзеркалення	Зміна орієнтації сцени	Покращення узагальнення на різні ракурси
Mosaic / змішування сцен	Поєднання кількох зображень у навчальному прикладі	Підвищення стійкості до складного фону
Розмиття	Імітація нечіткості зображення	Стійкість до менш якісних кадрів

Водночас аугментація не повинна спотворювати семантику класів. Наприклад, небезпечний предмет після перетворення повинен залишатися впізнаваним і зберігати правильну розмітку. Надмірні або неприродні зміни можуть призвести до того, що модель буде навчатися на прикладах, які не відповідають реальним умовам використання системи. Тому під час підготовки датасету доцільно використовувати ті перетворення, які наближені до реальних умов експлуатації системи та не порушують зміст зображення (рисунок 3.2).

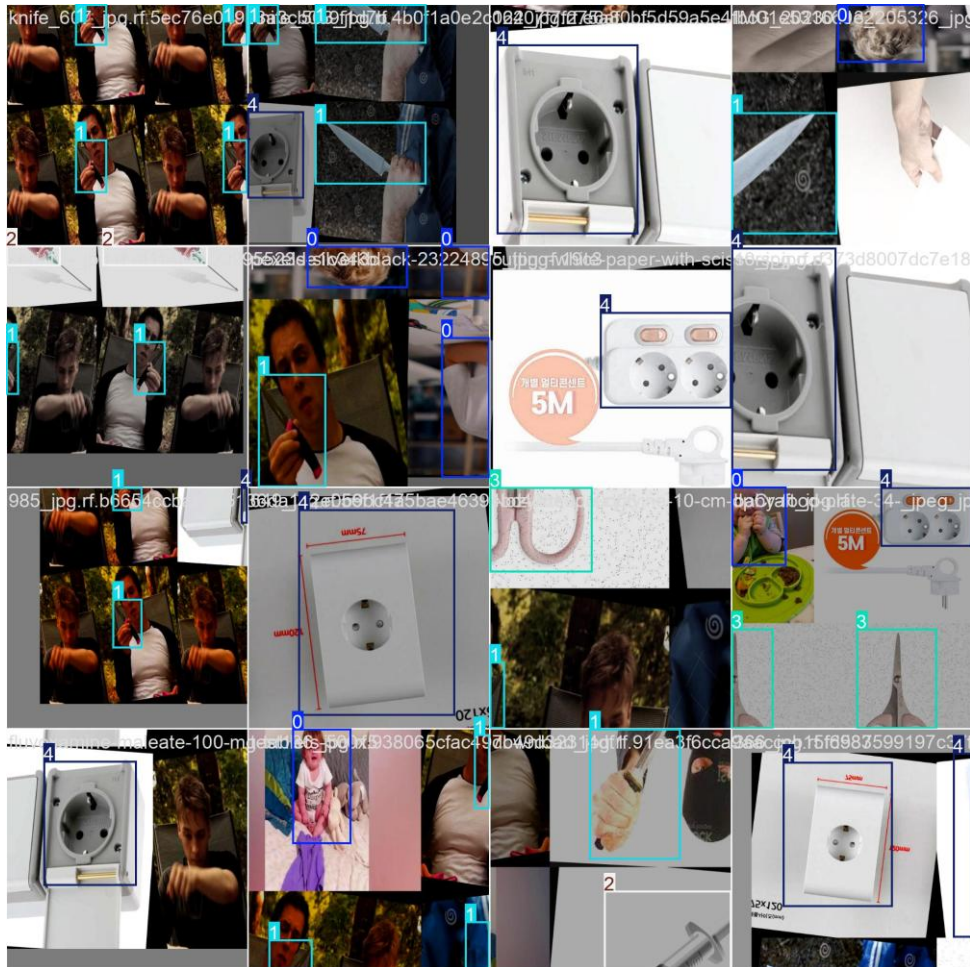


Рисунок 3.2 – Приклад навчального батчу після підготовки та аугментації даних

3.5 Налаштування середовища навчання

Навчання моделі виконувалося в хмарному середовищі Google Colab із використанням бібліотеки Ultralytics YOLO, яка надає інструменти для навчання, валідації та експорту моделей детекції об'єктів. Як базову модель було обрано YOLO11s, оскільки вона забезпечує прийнятний баланс між точністю, швидкістю та обчислювальною складністю для задачі детекції дитини й небезпечних предметів [20].

Вибір Google Colab зумовлений тим, що навчання нейронних мереж для задач комп'ютерного зору потребує значних обчислювальних ресурсів, зокрема використання графічного процесора. Локальне навчання на

пристрої з інтегрованою графікою є менш ефективним, оскільки виконання 30–40 епох на зображеннях розміром 640 пікселів може тривати значно довше та створювати надмірне навантаження на систему. Використання хмарного GPU-прискорення дозволило скоротити час навчання та стабільно провести експеримент.

Додатковою перевагою Google Colab є спрощене налаштування робочого середовища. У межах експерименту не потрібно було окремо встановлювати та узгоджувати локальні версії CUDA, драйверів, PyTorch та інших залежностей. У середовищі Colab було встановлено бібліотеку Ultralytics, підключено підготовлений набір даних, експортований із Roboflow у форматі YOLO, та збережено результати навчання: ваги моделі, графіки метрик, матриці помилок і приклади передбачень.

Середовище навчання було налаштовано для роботи з власним датасетом у форматі YOLO. У параметрах навчання було вказано шлях до файлу data.yaml, кількість епох, розмір зображення, розмір пакета та інші гіперпараметри. Навчання виконувалося на GPU, що дозволило прискорити процес оптимізації моделі порівняно з локальним запуском на пристрої без дискретної відеокарти.

Використання Google Drive для збереження датасету та результатів навчання забезпечило зручність організації експерименту. Після завершення навчання всі службові файли, графіки, ваги моделі та результати валідації залишалися доступними для подальшого аналізу. Це особливо важливо, оскільки отримані файли використовувалися у розділах, присвячених аналізу результатів.

У таблиці 3.3 наведено основні параметри навчання моделі. Вони були використані під час проведення експерименту в середовищі Google Colab і визначали ключові умови процесу оптимізації. Такий набір налаштувань забезпечив стабільність навчання та можливість подальшого аналізу результатів.

Таблиця 3.3 – Основні параметри навчання моделі

Параметр	Значення
Модель	yolo11s.pt
Кількість епох	40
Розмір зображення	640 × 640
Batch size	16
Patience	10
Optimizer	auto
Початкова швидкість навчання	0.01
Використання GPU	GPU (device 0)
Формат задачі	detect

3.6 Навчання моделі YOLO11s

Навчання моделі YOLO11s проводилося в середовищі Google Colab на підготовленому датасеті, експортованому з Roboflow у форматі YOLO. Для запуску експерименту було використано конфігураційний файл data.yaml, у якому визначено шляхи до навчальної, валідаційної та тестової вибірок, а також перелік класів об'єктів. Під час навчання модель отримувала зображення та відповідні анотації, після чого поступово коригувала ваги для підвищення точності локалізації й класифікації об'єктів [21].

У межах роботи було виконано основне навчання моделі на фінальній версії підготовленого датасету без окремого етапу донавчання. Такий підхід дозволив оцінити результати саме одного завершеного циклу навчання та надалі використовувати отримані ваги моделі для програмної реалізації системи.

У межах експерименту було виконано 40 епох навчання. Така кількість епох дала змогу простежити динаміку зміни втрат і метрик, а також оцінити, як змінюється якість моделі протягом навчання. Під час навчання автоматично зберігалися файли best.pt та last.pt. Файл best.pt

використано як основну модель для подальшої програмної реалізації системи.

Запуск навчання здійснювався засобами бібліотеки Ultralytics. У ньому було задано шлях до конфігураційного файлу датасету, кількість епох, розмір вхідного зображення, розмір пакета, параметр ранньої зупинки, пристрій для обчислень та директорію для збереження результатів експерименту (лістинг 3.1) [22].

Лістинг 3.1 – Фрагмент програмного коду запуску навчання моделі YOLO11s у середовищі Google Colab

```
from ultralytics import YOLO
model = YOLO("yolo11s.pt")
model.train(

data="/content/drive/MyDrive/child_safety/dataset/data.yam
l",

    epochs=40,
    imgsz=640,
    batch=16,
    patience=10,
    device=0,
    project="/content/drive/MyDrive/child_safety/results",
    name="yolo11s_v3_40epochs_batch16"
)
```

Наведений фрагмент коду демонструє основні параметри експерименту: використання моделі yolo11s.pt, навчання протягом 40 епох, розмір зображення 640×640 , розмір пакета 16, параметр patience зі значенням 10 та використання GPU. Такі налаштування відповідають задачі навчання моделі детекції об'єктів на власному датасеті та забезпечують баланс між якістю навчання й обчислювальними витратами [23].

Після завершення навчання було отримано набір службових файлів: графіки зміни втрат і метрик, матрицю помилок, криві Precision-Recall, приклади передбачень на валідаційній вибірці, а також ваги навченої моделі. Ці матеріали використано в четвертому розділі для аналізу результатів експериментальних досліджень.

3.7 Аналіз результатів навчання

Після завершення навчання моделі YOLO11s було отримано набір результатів, сформованих бібліотекою Ultralytics. До них належать графіки зміни втрат і метрик, матриця помилок, криві Precision-Recall, приклади передбачень на валідаційних зображеннях, а також ваги навченої моделі. Основним файлом для подальшої реалізації системи було обрано best.pt, оскільки він містить ваги моделі з найкращими показниками під час навчання. Використання саме цього файлу дозволяє застосовувати в системі не останній стан навчання, а варіант моделі, який показав найкращу якість на валідаційних даних.

Графіки навчання моделі дають змогу простежити загальну динаміку процесу оптимізації та оцінити стабільність її збіжності. Поступове зменшення значень функцій втрат свідчить про адаптацію моделі до підготовленого набору даних, тоді як зміна показників якості відображає покращення здатності моделі до виявлення та локалізації об'єктів (рисунок 3.3).

Отримані результати підтверджують, що навчання було виконано коректно, а модель сформувала ваги, придатні для подальшого використання в програмному модулі детекції. Підсумкові числові метрики та детальний аналіз якості роботи моделі розглянуто в четвертому розділі, щоб уникнути дублювання результатів експериментального оцінювання.

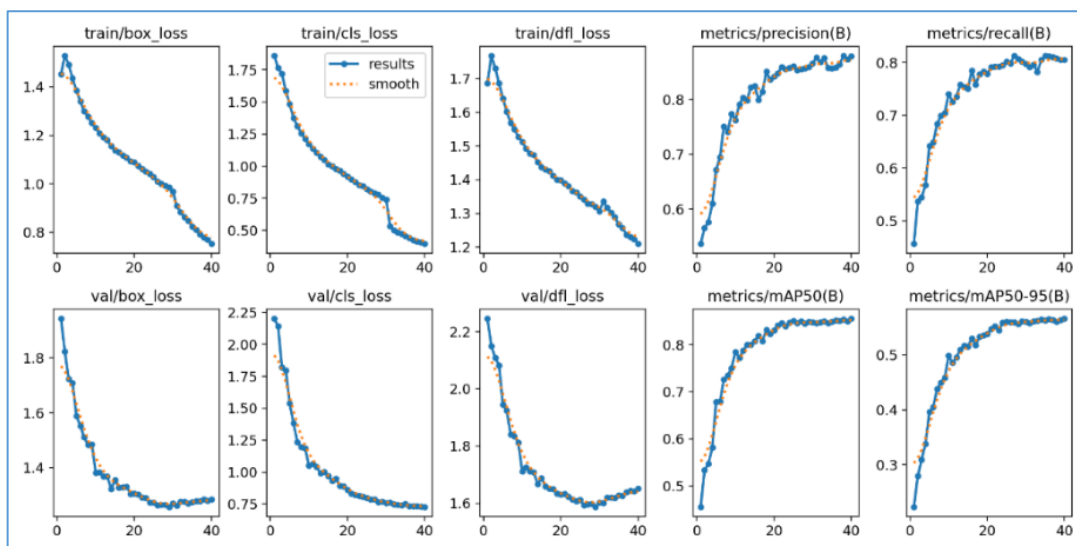


Рисунок 3.3 – Динаміка втрат і метрик під час навчання моделі YOLO11s

3.8 Реалізація програмного забезпечення засобами Python

Програмна реалізація системи виконувалася мовою Python. Для роботи з моделлю детекції використано бібліотеку Ultralytics, для оброблення зображень і нанесення графічних елементів – OpenCV, для математичних обчислень – стандартний модуль math. У програмі передбачено завантаження навченої моделі best.pt, оброблення вхідного зображення, отримання передбачень моделі та подальший розрахунок рівня небезпеки [24].

Основними класами небезпечних об'єктів у програмі є knife, medicine, scissors та socket. Пороги ризику задані в пікселях: високий ризик визначається за відстані не більше 120 px або за умови перетину рамок, середній – за відстані не більше 250 px, низький – якщо відстань перевищує 250 px. Такий підхід дозволяє не лише зафіксувати наявність небезпечного предмета в кадрі, а й оцінити його розташування відносно дитини (лістинг 3.2).

Лістинг 3.2 – Фрагмент програмного коду з визначенням небезпечних класів і порогів ризику

```
MODEL_PATH = "best.pt"
IMAGE_PATH = "test.jpg"
OUTPUT_PATH = "result.jpg"

DANGER_CLASSES = ["knife", "medicine", "scissors",
                  "socket"]

LOW_RISK_DISTANCE = 250
HIGH_RISK_DISTANCE = 120
```

Для оцінювання просторової близькості спочатку визначається центр обмежувальної рамки. Далі між центром рамки дитини та центром рамки небезпечного об'єкта обчислюється евклідова відстань. Окремо перевіряється факт перетину рамок, оскільки він може свідчити про близьке розташування об'єктів у межах зображення (лістинг 3.3).

Лістинг 3.3 – Функції визначення центра рамки, обчислення відстані та перевірки перетину рамок

```
def get_center(box):
    x1, y1, x2, y2 = box
    return int((x1 + x2) / 2), int((y1 + y2) / 2)

def calculate_distance(point1, point2):
    x1, y1 = point1
    x2, y2 = point2
    return math.sqrt((x2 - x1) ** 2 + (y2 - y1) ** 2)

def boxes_intersect(box1, box2):
    x1_min, y1_min, x1_max, y1_max = box1
    x2_min, y2_min, x2_max, y2_max = box2

    return not (x1_max < x2_min or x2_max < x1_min or
                y1_max < y2_min or y2_max < y1_min)
```

Функція визначення ризику перетворює числове значення відстані та логічний параметр перетину рамок у зрозумілий рівень небезпеки. Якщо рамки перетинаються, ситуація одразу класифікується як високий ризик. В інших випадках рішення приймається за встановленими порогами (лістинг 3.4).

Лістинг 3.4 – Функція визначення рівня ризику

```
def get_risk_level(distance, intersect=False):
    if intersect or distance <= HIGH_RISK_DISTANCE:
        return "HIGH"
    if distance <= LOW_RISK_DISTANCE:
        return "MEDIUM"
    return "LOW"
```

Після визначення рівня ризику програма формує результат: малює рамки об'єктів, центри рамок, лінії між дитиною та небезпечними предметами, підписує відстань і зберігає зображення з результатом. Якщо хоча б одна пара об'єктів має середній або високий ризик, система формує повідомлення про потенційно аномальний стан у межах виконаного аналізу.

Програмна реалізація також передбачає збереження проміжної інформації про кожну пару об'єктів. Для подальшого аналізу фіксується клас небезпечного предмета, рівень впевненості моделі, координати рамок, обчислена відстань, факт перетину та визначений рівень ризику. Це дає змогу не лише показати фінальний результат, а й пояснити, на основі яких даних система сформувала висновок.

Такий підхід підвищує прозорість роботи системи. Якщо користувач бачить повідомлення про аномальний стан, він може перевірити, який саме об'єкт був причиною спрацювання. Це важливо для практичного використання, оскільки система безпеки повинна бути не лише автоматичною, а й зрозумілою для людини.

Кодова структура системи побудована таким чином, щоб окремі функції відповідали окремим логічним операціям. Визначення центра рамки, обчислення відстані, перевірка перетину та визначення ризику реалізовані окремо. Це спрощує тестування програми й дозволяє змінювати окремі частини алгоритму без повного переписування коду.

Отже, програмна реалізація поєднує результати нейромережевої детекції з алгоритмічним аналізом просторового розташування об'єктів. Завдяки поділу коду на окремі функції система залишається зрозумілою, придатною для тестування та подальшого розширення.

3.9 Розроблення вебзастосунку на основі Streamlit

Для зручної взаємодії користувача із системою було розроблено вебзастосунок на основі Streamlit. Цей інструмент дозволяє швидко створювати інтерактивні застосунки мовою Python без необхідності розроблення складного фронтенду. У створеному інтерфейсі користувач може завантажити зображення, налаштувати поріг впевненості моделі та переглянути результат аналізу [25].

Одним із основних елементів інтерфейсу є блок завантаження зображення. Користувач може обрати файл у форматі jpg, jpeg або png, після чого зображення передається до моделі детекції. Також у вебзастосунку реалізовано слайдер для налаштування порогу впевненості моделі. Це дозволяє змінювати чутливість детекції без редагування програмного коду (лістинг 3.5).

Лістинг 3.5 – Фрагмент програмного коду для завантаження зображення та налаштування порогу впевненості

```
uploaded_file = st.file_uploader(  
    "Оберіть зображення",  
    type=["jpg", "jpeg", "png"]  
)
```

Продовження лістингу 3.5

```
confidence = st.slider(
    "Поріг впевненості моделі",
    min_value=0.05,
    max_value=0.90,
    value=0.25,
    step=0.05
)
```

Після завантаження зображення система виконує детекцію об'єктів за допомогою навченої моделі YOLO11s. Далі результати передаються до алгоритму оцінювання ризику, який визначає центри обмежувальних рамок, обчислює відстані між дитиною та небезпечними предметами, перевіряє перетин рамок і встановлює рівень ризику для кожної пари об'єктів.

Результати аналізу подаються у візуальній і табличній формах. Користувач бачить оброблене зображення з рамками, підписами класів, центрами об'єктів і лініями відстані. Додатково виводяться кількісні показники: кількість виявлених дітей, кількість небезпечних об'єктів і кількість оцінених пар ризику. Результати також організовано у вигляді вкладок: результат детекції, оригінальне зображення та таблиця аналізу ризику. У таблиці зазначається назва небезпечного об'єкта, рівень впевненості моделі, відстань до дитини, факт перетину рамок і визначений рівень ризику. Якщо хоча б для однієї пари «дитина – небезпечний об'єкт» визначено середній або високий рівень ризику, система формує повідомлення про потенційно аномальний стан у межах виконаного аналізу.

Таким чином, вебзастосунок на основі Streamlit забезпечує зручний спосіб використання навченої моделі та алгоритму оцінювання ризику. Він дозволяє перейти від окремої моделі детекції до інтерактивної інтелектуальної системи, яка приймає зображення, виконує аналіз і подає результат у зрозумілій для користувача формі.

4 ЕКСПЕРИМЕНТАЛЬНІ ДОСЛІДЖЕННЯ ТА АНАЛІЗ РЕЗУЛЬТАТІВ

4.1 Методика проведення експериментальних досліджень

Експериментальні дослідження виконувалися з метою перевірки якості навченої моделі YOLO11s та оцінювання роботи розробленої інтелектуальної системи в задачі виявлення аномальних станів ігрового середовища дитини. Оскільки система складається з двох основних частин – моделі детекції об'єктів і алгоритму оцінювання рівня небезпеки, експериментальна перевірка охоплювала як результати розпізнавання об'єктів, так і здатність системи формувати висновок про потенційно ризиковий стан у межах аналізованого зображення.

У межах роботи було виконано основне навчання моделі YOLO11s на фінально сформованому наборі даних. Датасет було підготовлено в Roboflow, експортовано у форматі YOLO та використано для навчання у середовищі Google Colab. Модель навчалася протягом 40 епох із розміром вхідного зображення 640×640 пікселів та розміром пакета 16 зображень. Після завершення навчання було отримано ваги моделі, графіки зміни втрат і метрик, матрицю помилок, PR-криві та приклади передбачень на валідаційних зображеннях.

Оцінювання моделі проводилося за результатами, отриманими на валідаційній вибірці. Для цього аналізувалися стандартні метрики детекції об'єктів, зокрема Precision, Recall, mAP50 та mAP50-95. Такі метрики дозволяють оцінити модель з різних сторін: точність правильних спрацювань, здатність знаходити наявні об'єкти та якість локалізації обмежувальних рамок. Особливу увагу було приділено метриці mAP50-95, оскільки вона є суворішою та краще відображає якість побудови рамок для об'єктів різного масштабу.

Методика експерименту передбачала послідовне виконання кількох етапів: аналіз підсумкових метрик моделі, оцінювання якості розпізнавання окремих класів, виявлення типових помилок, демонстрацію роботи програмного модуля та оцінювання практичної придатності системи. Окремо перевірялося, чи може система не лише виявити дитину та небезпечні предмети на зображенні, а й виконати додатковий аналіз їхнього взаємного розташування.

Під час експериментальної перевірки враховувалося, що результати роботи системи залежать не тільки від правильності класифікації об'єктів, а й від точності побудови обмежувальних рамок. Це є важливим для даної роботи, оскільки координати рамок надалі використовуються для визначення центрів об'єктів, обчислення піксельної відстані та перевірки перетину рамок. Отже, якість первинної детекції безпосередньо впливає на подальше оцінювання рівня ризику.

4.2 Характеристика тестової вибірки

Для оцінювання моделі використовувалися зображення, що не застосовувалися для безпосереднього оновлення ваг під час навчання. Тестові та валідаційні приклади були сформовані в межах підготовленого датасету й містили сцени з дитиною та потенційно небезпечними предметами. До цільових класів належали: *baby*, *knife*, *medicine*, *scissors* та *socket*.

Використання окремих прикладів для перевірки є важливим, оскільки воно дозволяє оцінити здатність моделі працювати не лише з навчальними зображеннями, а й з даними, які не брали участі в оновленні її ваг. Це зменшує ризик переоцінювання якості моделі та дає змогу краще оцінити її здатність узагальнювати ознаки об'єктів на нових сценах.

Особливістю тестових прикладів була наявність складних умов для комп'ютерного зору: різний масштаб об'єктів, часткове перекриття,

неоднорідний фон, зміни освітлення та відмінні ракурси зйомки. Такі умови є важливими для перевірки практичної придатності системи, оскільки в реальному побутовому середовищі небезпечні предмети можуть бути невеликими, частково закритими іншими об'єктами або розташованими на фоні, схожому за кольором і текстурою.

Під час тестування враховувалося не лише те, чи правильно модель визначає клас об'єкта, а й наскільки коректно побудована обмежувальна рамка. Це важливо, оскільки подальший модуль оцінювання небезпеки використовує координати рамок для обчислення центрів об'єктів, визначення піксельної відстані між ними та перевірки перетину рамок. Отже, якість локалізації безпосередньо впливає на результат подальшого аналізу ризику.

4.3 Оцінювання ефективності моделі за метриками Precision, Recall, mAP50 та mAP50-95

Оцінювання якості моделі виконувалося за стандартними метриками детекції об'єктів. Вони дозволяють проаналізувати не лише здатність моделі правильно визначати класи об'єктів, а й якість побудови обмежувальних рамок, що є важливим для подальшого оцінювання ризику [26].

Для аналізу використовувалися такі метрики:

- Precision. Характеризує частку правильних спрацювань серед усіх передбачень моделі;
- Recall. Показує повноту виявлення реальних об'єктів;
- mAP50. Оцінює середню точність за порогу IoU 0,5;
- mAP50-95. Є суворішою метрикою, оскільки враховує якість локалізації на кількох порогах IoU від 0,5 до 0,95.

Такий набір метрик дозволяє оцінити модель з різних сторін. Precision показує, наскільки надійними є передбачення моделі, Recall відображає здатність не пропускати наявні об'єкти, а mAP-метрики характеризують

загальну якість детекції та локалізації. Для розробленої системи це особливо важливо, оскільки координати обмежувальних рамок надалі використовуються для обчислення центрів об'єктів, піксельної відстані та перевірки перетину рамок. Підсумкові метрики моделі після 40 епох навчання наведено в таблиці 4.1.

Таблиця 4.1 – Підсумкові метрики моделі після 40 епох навчання

Метрика	Значення
Precision	0,8799
Recall	0,8052
mAP50	0,8557
mAP50-95	0,5664

Отримані значення свідчать про достатню якість моделі для задачі виявлення дитини та небезпечних об'єктів у межах підготовленого набору даних. Найвищим серед підсумкових показників є Precision, що вказує на відносно невелику кількість хибних спрацювань. Це означає, що більшість об'єктів, які модель позначає як виявлені, справді відповідають цільовим класам.

Показник Recall є нижчим, тому частина складних або малих об'єктів може залишатися невиявленою. Для задачі безпеки це є важливим обмеженням, оскільки пропуск небезпечного предмета може вплинути на подальше формування висновку про ризик. Значення mAP50-95 є очікувано нижчим за mAP50, оскільки ця метрика суворіше оцінює точність локалізації обмежувальних рамок.

Таким чином, модель демонструє достатньо стабільні результати для використання в прототипі інтелектуальної системи. Водночас нижчі значення Recall та mAP50-95 показують, що подальше покращення моделі

доцільно спрямувати на точніше виявлення дрібних і складних об'єктів, а також на підвищення якості локалізації рамок.

4.4 Аналіз результатів детекції дитини та небезпечних об'єктів

Для детальнішого аналізу результатів було розглянуто нормалізовану матрицю помилок, сформовану після навчання моделі. Вона дозволяє оцінити, наскільки стабільно модель розпізнає кожен із класів, а також визначити, які об'єкти частіше пропускаються або помилково відносяться до фону.

За результатами матриці помилок найстабільніше модель розпізнає класи *medicine*, *scissors* та *baby*. Для класу *medicine* частка правильного розпізнавання становить 0,92, для *scissors* – 0,91, а для *baby* – 0,89. Це свідчить про те, що модель достатньо добре засвоїла візуальні ознаки цих об'єктів. Такий результат можна пояснити відносно помітними ознаками класів, а також достатньою кількістю прикладів у навчальному наборі.

Складнішими для моделі виявилися класи *knife* та *socket*. Для класу *knife* частка правильного розпізнавання становить 0,75, а для *socket* – 0,77. Такі результати можуть бути пов'язані з малим розміром об'єктів, різними ракурсами, слабким контрастом із фоном або частковим перекриттям. Крім того, для цих класів помітною є частка пропусків, коли об'єкт відноситься до фону, що особливо важливо враховувати під час подальшого аналізу безпеки (рисунок 4.1).

Аналіз матриці також показує, що для проблемних класів помилки частіше пов'язані не з плутанням між цільовими категоріями, а з віднесенням об'єкта до фону. Це означає, що для класів *knife* та *socket* основною проблемою є саме пропуск об'єкта, а не його неправильна класифікація як іншого небезпечного предмета.

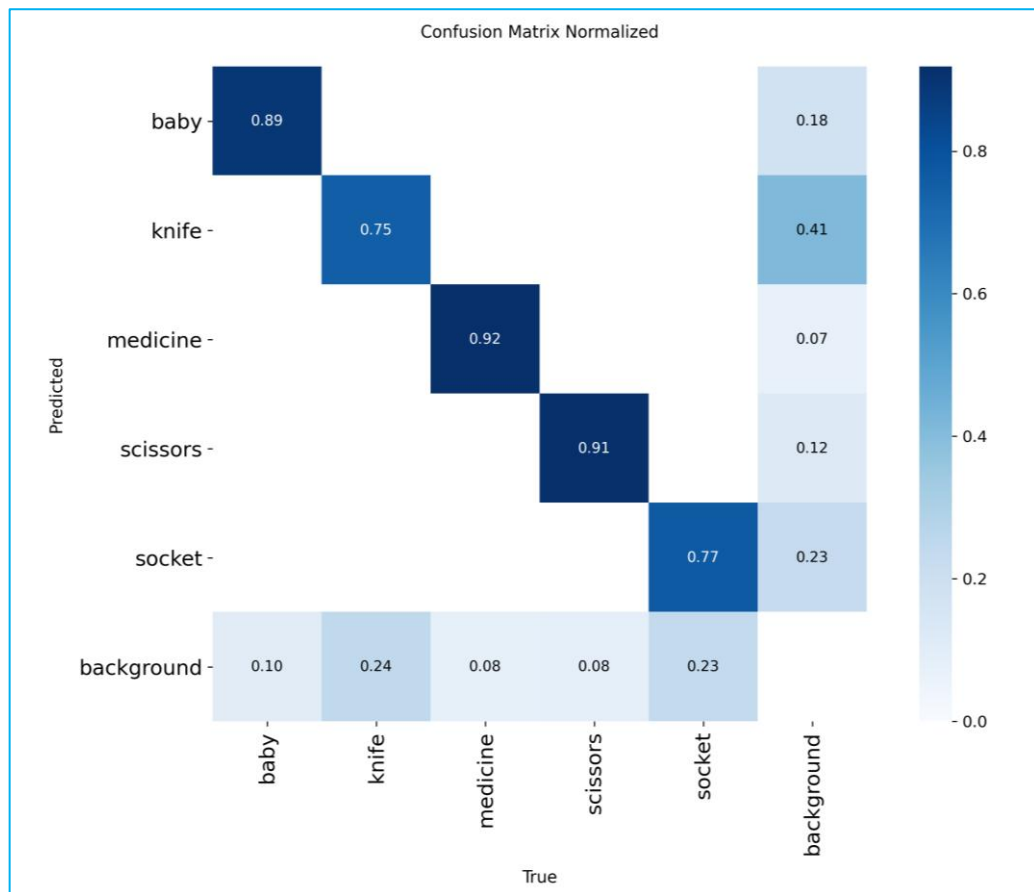


Рисунок 4.1 – Нормалізована матриця помилок моделі YOLO11s

Результати перевірки на валідаційних зображеннях показують, що модель здатна виявляти цільові класи в різних візуальних умовах. У наборі прикладів присутні сцени з різним фоном, масштабом об'єктів, положенням дитини в кадрі та ступенем помітності небезпечних предметів. Це дозволяє оцінити, наскільки модель узагальнює ознаки, отримані під час навчання, та чи може застосовувати їх до зображень, які не використовувалися для безпосереднього оновлення ваг.

Особливо важливим є те, що модель може одночасно знаходити дитину та потенційно небезпечні предмети в межах однієї сцени. Для розробленої системи це має принципове значення, оскільки подальший аналіз ризику базується не лише на факті появи небезпечного об'єкта, а й на його розташуванні відносно дитини в межах зображення. Отримані

обмежувальні рамки, класи та значення впевненості надалі використовуються для визначення рівня небезпеки.

Під час візуального аналізу результатів важливо звертати увагу не лише на правильність визначення класу, а й на якість побудови обмежувальної рамки. Для звичайної задачі класифікації достатньо правильно визначити об'єкт, однак у цій роботі координати рамки використовуються для подальших геометричних розрахунків. Тому навіть частково неточна детекція може впливати на результат обчислення піксельної відстані між дитиною та небезпечним предметом.

Найбільш стабільними є випадки, коли дитина займає помітну частину кадру і має достатній контраст із фоном. Небезпечні предмети розпізнаються складніше, особливо якщо вони дрібні, мають незвичний ракурс або розташовані поруч із великою кількістю інших об'єктів. Це підтверджує необхідність подальшого розширення датасету прикладами зі складними побутовими сценами, малими об'єктами та різними умовами освітлення.

На прикладах детекції на валідаційних зображеннях видно, що модель здатна виявляти дитину та окремі небезпечні предмети в різних сценах. Окрему увагу приділено не лише наявності обмежувальних рамок, а й тому, наскільки повно вони охоплюють об'єкт, оскільки координати рамок надалі використовуються для розрахунку відстані між дитиною та небезпечним предметом. Водночас частина передбачень має нижчий рівень впевненості, що підтверджує складність задачі для дрібних або частково помітних об'єктів (рисунки 4.2).

Отримані результати демонструють, що модель може бути використана як основа для модуля оцінювання ризику, однак якість кінцевого висновку залежить від точності первинної детекції. Якщо небезпечний предмет не буде виявлений або рамка буде побудована неточно, система може некоректно оцінити його розташування відносно дитини.

для обчислення піксельної відстані між дитиною та небезпечним предметом. Якщо рамка зміщена, розрахована відстань може змінитися, що вплине на визначення рівня ризику.

Хибні спрацювання можуть виникати тоді, коли безпечні предмети мають форму, схожу на небезпечний об'єкт. Наприклад, видовжені предмети можуть бути схожими на ніж, а невеликі елементи інтер'єру – на розетку. Такі випадки не є критичними для етапу навчання, але в реальному використанні можуть призводити до зайвих попереджень.

Окремою причиною помилок може бути неоднорідність самого набору даних. Якщо одні класи представлені більшою кількістю зображень або мають більш різноманітні приклади, модель засвоює їхні ознаки краще. Класи з меншою кількістю прикладів або з великою візуальною варіативністю можуть розпізнаватися нестабільніше.

Для задачі безпеки дитини більш критичними є помилки пропуску небезпечного предмета, ніж хибні спрацювання. Хибне спрацювання може створити зайве попередження, але пропуск ножа, ліків або розетки може призвести до того, що система не зафіксує потенційно ризиковий об'єкт у кадрі. Тому при подальшому розвитку системи доцільно приділити особливу увагу підвищенню Recall для складних класів.

У подальшому зменшення кількості таких помилок можливе шляхом розширення датасету складними прикладами, додаванням більшої кількості зображень із дрібними предметами, різними ракурсами, частковими перекриттями та складним побутовим фоном. Це дозволить підвищити стійкість моделі до реальних умов використання, де об'єкти можуть бути частково закритими, розмитими або розташованими на фоні, схожому за кольором і текстурою.

Ще одним напрямом удосконалення є використання відеопотоку замість одиничних зображень. Аналіз кількох послідовних кадрів дозволив би зменшити вплив випадкових помилок окремої детекції. Якщо об'єкт не був виявлений в одному кадрі, але стабільно визначається в наступних,

система могла б формувати більш надійний висновок про стан середовища в межах аналізованої сцени.

Крім того, робота з відеопотоком дала б змогу враховувати не лише поточне положення об'єктів, а й зміну їхнього розташування в часі. Наприклад, система могла б фіксувати ситуації, коли відстань між дитиною та небезпечним предметом у кадрі поступово зменшується. Такий підхід підвищив би превентивний характер системи, оскільки попередження могло б формуватися не лише за результатом одного зображення, а й на основі послідовного аналізу кількох кадрів.

4.6 Аналіз динаміки навчання моделі протягом 40 епох

Аналіз динаміки навчання моделі YOLO11s протягом 40 епох показує, що обрана тривалість експерименту була достатньою для отримання стабільних результатів детекції в межах підготовленого набору даних. Оскільки графіки зміни втрат і метрик уже наведено в розділі 3, у цьому підрозділі акцент зроблено на узагальненні отриманої динаміки та її значенні для практичної роботи системи.

Поступове зменшення `train/box_loss`, `train/cls_loss` і `train/dfl_loss` свідчить про коректний процес оптимізації моделі. У практичному сенсі це означає, що модель поступово покращувала точність побудови обмежувальних рамок, класифікацію об'єктів і уточнення координат їхніх меж. Водночас зростання Precision, Recall, mAP50 та mAP50-95 показує, що якість детекції покращувалася не лише на рівні навчальних даних, а й на валідаційних прикладах.

Важливим результатом є те, що протягом навчання не спостерігалось різкого погіршення метрик на валідаційній вибірці. Це свідчить про те, що модель не лише адаптувалася до навчальних прикладів, а й зберігала здатність узагальнювати ознаки об'єктів на відокремлених даних. Такий результат є важливим для подальшого використання моделі в системі

оцінювання ризику, оскільки якість детекції безпосередньо впливає на правильність побудови рамок і подальші геометричні розрахунки.

Разом з тим аналіз динаміки навчання не означає, що модель повністю позбавлена помилок. Найбільш складними залишаються випадки з малими об'єктами, частковим перекриттям і низьким контрастом із фоном. Тому подальше покращення результатів доцільно пов'язувати не лише зі збільшенням кількості епох, а насамперед із розширенням і збалансуванням датасету, додаванням складних побутових сцен та збільшенням кількості прикладів для класів *knife*, *socket* і *medicine*.

Таким чином, аналіз динаміки навчання показує, що модель досягла прийняттого рівня стабільності для використання в розробленому прототипі. Водночас подальше вдосконалення системи має бути пов'язане передусім із покращенням якості та різноманітності навчальних даних, оскільки саме вони значною мірою визначають здатність моделі працювати в реальних побутових умовах.

4.7 Демонстрація роботи системи на тестових зображеннях

Для демонстрації роботи інтелектуальної системи було використано вебзастосунок, розроблений на основі Streamlit. Користувач завантажує тестове зображення, після чого система виконує детекцію об'єктів за допомогою навченої моделі YOLO11s. Далі результати детекції передаються до алгоритму оцінювання ризику, який аналізує пари «дитина – небезпечний об'єкт».

Під час демонстрації система відображає оброблене зображення з обмежувальними рамками та підписами класів. Для дитини й небезпечних предметів визначаються центри рамок, після чого між ними можуть бути побудовані лінії, що показують піксельну відстань і взаємне розташування об'єктів у межах зображення. Такий спосіб візуалізації дозволяє швидко

зрозуміти, на основі яких даних система сформувала висновок про стан сцени.

Якщо відстань між дитиною та небезпечним предметом відповідає середньому або високому рівню ризику, система формує повідомлення про потенційно аномальний стан у межах виконаного аналізу. У цьому випадку результат не обмежується лише текстовим повідомленням: користувач бачить також конкретний об'єкт, який став причиною спрацювання, його клас, рівень впевненості моделі та розраховану піксельну відстань.

Додатково результати подаються у вигляді таблиці аналізу ризику. Такий формат є зручним для експериментальної перевірки, оскільки дозволяє порівняти кілька пар об'єктів в одному зображенні. Якщо в кадрі присутні кілька небезпечних предметів, система окремо оцінює кожну пару та формує загальний висновок на основі найвищого визначеного рівня ризику. При цьому такий підхід не потребує ручного вибору об'єкта для аналізу, оскільки всі знайдені небезпечні предмети автоматично порівнюються з координатами виявленої дитини.

На екрані результату користувач бачить загальний статус сцени, кількість виявлених дітей, кількість небезпечних об'єктів та кількість оцінених пар ризику (рисунки 4.3). Такий підхід дозволяє швидко оцінити загальну ситуацію на зображенні ще до детального перегляду результатів. Якщо система визначає середній або високий рівень ризику хоча б для однієї пари об'єктів, у верхній частині інтерфейсу відображається повідомлення про потенційно аномальний стан.

Окрему роль відіграє таблиця аналізу ризику. У ній наведено назву небезпечного об'єкта, рівень впевненості моделі, піксельну відстань до дитини, факт перетину рамок і визначений рівень ризику. Завдяки цьому користувач може зрозуміти, який саме об'єкт став причиною спрацювання системи та на основі яких параметрів було сформовано висновок. Це підвищує прозорість роботи вебзастосунку й полегшує аналіз результатів під час експериментальної перевірки (рисунки 4.3).

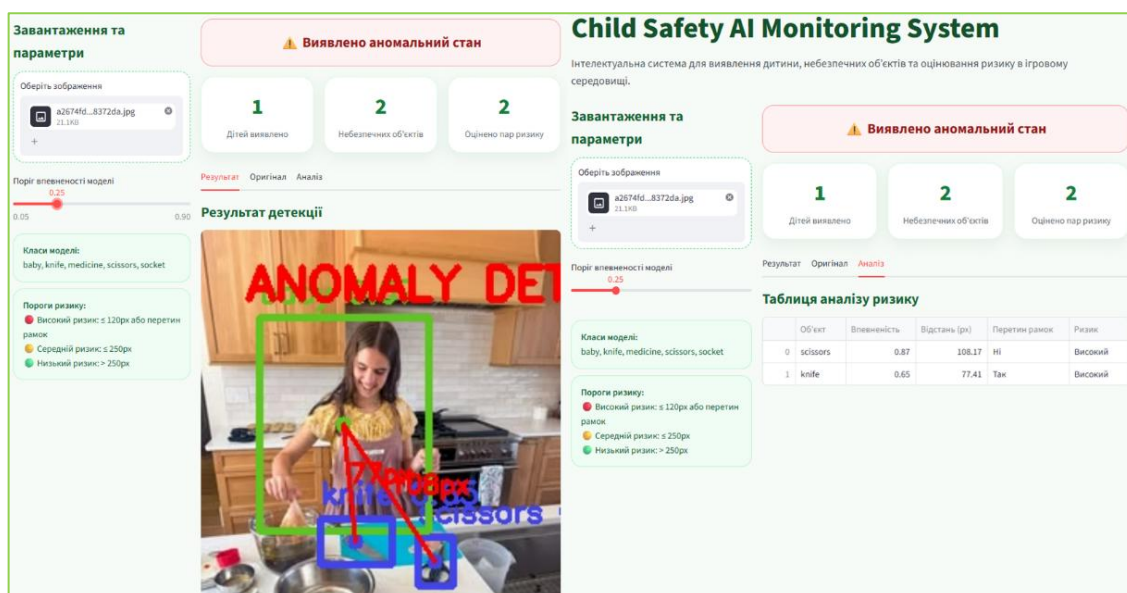


Рисунок 4.3 – Приклад роботи вебзастосунку з виявленням потенційно аномального стану та таблицею аналізу ризику

Отже, демонстрація на тестових прикладах підтверджує, що розроблена система здатна поєднувати результати детекції з алгоритмом оцінювання ризику та подавати результат у зрозумілій для користувача формі. Це дозволяє використовувати вебзастосунок як інструмент для перевірки роботи моделі й аналізу потенційно ризикових ситуацій на статичних зображеннях.

4.8 Оцінювання практичної придатності розробленої системи

Розроблена система може розглядатися як прототип інтелектуального засобу моніторингу безпеки дитини в ігровому середовищі. Вона демонструє можливість поєднання моделі комп'ютерного зору з алгоритмом геометричного аналізу для формування висновку про потенційно ризикові ситуації на основі зображення.

Основною перевагою системи є її зрозуміла логіка роботи. На відміну від повністю закритих моделей, результат яких складно інтерпретувати,

запропонована система показує користувачу не лише підсумковий статус, а й проміжні дані: клас об'єкта, координати рамки, піксельну відстань до дитини та визначений рівень ризику. Це підвищує прозорість роботи системи та полегшує аналіз її помилок.

Водночас розроблена система має певні обмеження. Насамперед оцінювання відстані виконується у пікселях, тому воно залежить від масштабу зображення, положення камери та перспективи. Система не визначає реальну фізичну відстань у сантиметрах або метрах, а лише оцінює взаємне розташування об'єктів у площині кадру. Це обмеження є прийнятним для прототипу, але потребує врахування під час подальшого розвитку.

Слід також зазначити, що самі порогові значення піксельної відстані, за якими визначається рівень ризику, було обрано експериментально з урахуванням розміру тестових зображень, що використовувалися під час розробки та демонстрації прототипу. Ці значення не нормалізовані відносно розміру обмежувальної рамки дитини чи роздільної здатності вхідного зображення, тому при завантаженні зображень із істотно іншим масштабом ці самі порогові значення можуть некоректно відображати реальну просторову близькість об'єктів: однаковій фізичній відстані на зображенні більшого розміру відповідатиме більша піксельна відстань, що може призвести до заниження визначеного рівня ризику. Для практичного використання системи з довільними вхідними зображеннями доцільним удосконаленням є нормалізація порогових значень відносно розміру обмежувальної рамки дитини або масштабування зображення до фіксованого розміру перед обчисленням відстані.

Окремої уваги потребує асиметрія наслідків помилок першого та другого роду для задачі безпеки дитини. Отримане значення Recall (0,8052) означає, що приблизно кожен п'ятий небезпечний об'єкт у тестовій вибірці модель не виявляє. Для звичайної задачі класифікації це прийнятний показник, проте для системи попередження про небезпеку пропуск об'єкта

є критичнішим за хибне спрацювання: хибна тривога лише викликає зайву перевірку зображення користувачем, тоді як пропущений ніж чи розетка поруч із дитиною означає відсутність попередження саме в той момент, коли воно потрібне. Це означає, що для практичного застосування системи доцільніше орієнтуватися на нижче значення порогу впевненості моделі, навіть якщо це збільшить кількість хибних спрацювань, оскільки в задачах безпеки вартість пропущеної небезпеки вища за вартість зайвого попередження.

Практична придатність системи залежить не лише від архітектури моделі, а й від обраного користувачем порогу впевненості детекції, який у розробленому вебзастосунку можна налаштовувати в інтерфейсі. Зниження порогу підвищує ймовірність виявлення дрібних або частково перекритих об'єктів класів *knife* і *socket*, для яких точність розпізнавання є нижчою (0,75 та 0,77 відповідно), однак одночасно збільшує кількість хибних спрацювань. Для практичного використання системи як засобу попереднього контролю доцільнішим є помірно знижений порог впевненості, що зменшує ризик пропуску небезпечного об'єкта за рахунок прийнятного збільшення кількості повідомлень, які потребують додаткової перевірки користувачем.

Ще одним обмеженням є робота зі статичними зображеннями. У практичних умовах безпека дитини часто пов'язана з рухом, тому наступним етапом розвитку може бути оброблення відеопотоку. Це дозволило б аналізувати не лише поточне розташування об'єктів, а й зміну їхнього положення в часі. Наприклад, система могла б фіксувати ситуації, коли піксельна відстань між дитиною та небезпечним предметом у кадрі поступово зменшується.

Перспективним напрямом удосконалення є розширення набору класів. У межах цієї роботи розглядалися п'ять основних класів, однак у реальному середовищі потенційно небезпечними можуть бути також побутова хімія, дрібні деталі, гарячі предмети, кабелі, скляні предмети та

інші джерела ризику. Додавання нових класів дозволило б зробити систему більш універсальною.

Практична придатність системи також полягає в тому, що її логіка може бути адаптована до різних сценаріїв використання. Наприклад, подібний підхід можна застосовувати не лише для аналізу дитячої кімнати, а й для інших контрольованих просторів, де необхідно виявляти потенційно ризикове розташування людини відносно певних об'єктів. При цьому основна структура системи залишається незмінною: модель детекції визначає об'єкти, а додатковий алгоритм аналізує їхнє взаємне розташування та формує висновок про рівень ризику.

Варто також зазначити, що перевірка практичної придатності виконувалася на зображеннях із підготовленого набору даних, сформованого шляхом об'єднання кількох публічних датасетів і самостійно доданих складних сцен через платформу Roboflow. Тестова вибірка є частиною цього ж комбінованого набору, а не окремо зібраною колекцією фотографій реальних дитячих кімнат користувачів. Це означає, що отримані показники точності характеризують роботу системи в межах розподілу даних, на якому її було навчено, але не підтверджують напряду її стійкості до зображень із камер, типів приміщень або ракурсів, які не були представлені під час формування датасету. Перевірка на незалежно зібраних фотографіях реальних побутових середовищ є необхідним наступним кроком для остаточного підтвердження практичної придатності системи.

Таким чином, розроблена інтелектуальна система є придатною для демонстрації принципу автоматичного виявлення потенційно аномальних станів ігрового середовища на основі аналізу зображення. Вона не замінює нагляд дорослих, але може бути використана як додатковий інструмент попереднього контролю, який допомагає звернути увагу користувача на можливі ризикові ситуації.

ВИСНОВКИ

У ході виконання кваліфікаційної роботи було розглянуто задачу розроблення інтелектуальної системи детекції аномальних станів ігрового середовища для превентивного забезпечення безпеки дитини. Актуальність теми пов'язана з необхідністю своєчасного виявлення потенційно небезпечних об'єктів у дитячому середовищі. При цьому небезпека визначається не лише фактом наявності такого об'єкта в кадрі, а й його просторовим розташуванням відносно дитини.

У першому розділі було проаналізовано предметну область, особливості ігрового середовища дитини, основні джерела небезпеки та проблеми забезпечення безпеки дітей у побутовому просторі. Розглянуто сучасні підходи до моніторингу безпеки, принципи роботи систем комп'ютерного зору, методи виявлення та класифікації об'єктів, а також можливості застосування нейронних мереж у задачах безпеки. У результаті було обґрунтовано доцільність використання методів глибинного навчання та моделей сімейства YOLO для автоматизованої детекції дитини й потенційно небезпечних предметів.

У другому розділі було сформовано вимоги до інтелектуальної системи та визначено її основні функціональні компоненти. Система повинна приймати зображення ігрового або побутового середовища, виконувати детекцію дитини та небезпечних об'єктів, визначати координати обмежувальних рамок, обчислювати просторову близькість між об'єктами та формувати висновок про рівень ризику. Запропонована архітектура має модульну структуру, що дозволяє відокремити етапи введення даних, детекції об'єктів, структуризації результатів, оцінювання ризику, виявлення аномального стану та відображення результатів користувачу.

У межах роботи було визначено цільові класи об'єктів для детекції: дитина, ніж, ліки, ножиці та розетка. Такий вибір класів відповідає

поставленій задачі, оскільки зазначені об'єкти є поширеними у побутовому середовищі та можуть становити потенційну небезпеку для дитини. Для оцінювання рівня ризику було використано підхід, заснований на аналізі просторової близькості між центрами обмежувальних рамок дитини та небезпечного об'єкта. На основі евклідової відстані та заданих порогових значень система формує висновок про низький, середній або високий рівень небезпеки.

У третьому розділі було виконано формування та підготовку набору даних для навчання моделі детекції об'єктів. Датасет було підготовлено з урахуванням специфіки поставленої задачі, перевірено коректність анотацій, уніфіковано класи та застосовано аугментацію даних для підвищення стійкості моделі до змін освітлення, масштабу, положення об'єктів і складності фону. Для навчання було обрано модель YOLO11s, яка забезпечує прийнятний баланс між точністю, швидкодією та обчислювальною ефективністю. Навчання моделі виконувалося в середовищі Google Colab із використанням бібліотеки Ultralytics YOLO.

Практична реалізація системи була виконана засобами мови програмування Python із використанням моделі YOLO11s, бібліотек для оброблення зображень та вебфреймворку Streamlit. Розроблений вебзастосунок дозволяє користувачу завантажувати зображення, запускати детекцію, переглядати знайдені об'єкти, оцінювати відстань між дитиною та небезпечними предметами, а також отримувати структурований висновок про наявність або відсутність потенційно аномального стану.

У четвертому розділі було проведено експериментальне дослідження роботи навченої моделі та розробленої системи. Аналіз результатів показав, що модель здатна виявляти дитину та основні класи потенційно небезпечних об'єктів у межах тестових зображень. Було встановлено, що якість детекції залежить від складності сцени, масштабу об'єкта, освітлення, часткових перекриттів і візуальної подібності окремих предметів до фону. Найбільш складними для моделі є випадки, коли небезпечний предмет має

малий розмір, частково перекритий іншими об'єктами або розташований у візуально насиченому середовищі.

Отримані результати підтвердили працездатність запропонованого підходу. Поєднання нейронної детекції об'єктів із геометричним аналізом просторової близькості дозволяє не лише виявляти об'єкти на зображенні, а й формувати інтерпретований висновок про потенційний ризик. Такий підхід є важливим для задач превентивного забезпечення безпеки, оскільки користувач отримує не просто зображення з обмежувальними рамками, а пояснюваний результат: який об'єкт виявлено, на якій відстані він розташований від дитини та який рівень небезпеки визначено системою.

Разом із тим розроблена система має певні обмеження. Оцінювання відстані виконується у пікселях, тому воно не є повним еквівалентом реальної фізичної відстані між об'єктами. Результат може залежати від ракурсу камери, масштабу зображення, перспективи та якості вхідних даних. Крім того, система не замінює нагляд дорослих, а може розглядатися як допоміжний інструмент для привернення уваги до потенційно небезпечних ситуацій.

Отже, мету кваліфікаційної роботи досягнуто. У роботі було розроблено концепцію та програмний прототип інтелектуальної системи, яка виконує детекцію дитини й потенційно небезпечних предметів, аналізує їхнє взаємне розташування та формує висновок про рівень ризику. Практичне значення отриманих результатів полягає в можливості використання запропонованого підходу як основи для подальшого розвитку систем комп'ютерного зору, орієнтованих на превентивне забезпечення безпеки дитини. Подальше вдосконалення системи може бути пов'язане з розширенням кількості класів небезпечних об'єктів, адаптацією до відеопотоку в режимі реального часу, використанням калібрування камери для точнішого оцінювання відстані та підвищенням стійкості моделі до складних побутових умов.

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. Стахевич О. В., Гриньова О. Є. Комп'ютерний зір для контролю безпеки ігрового середовища дитини. *Радіоелектроніка та молодь у XXI столітті* : матеріали 30-го Міжнар. молодіж. форуму, 22-24 квітня 2026 р. Харків : ХНУРЕ, 2025. Т. 6. С. 120–122.
2. World report on child injury prevention / ed. by P. M. M, Unicef, W. H. Organization. Geneva, Switzerland : World Health Organization, 2008. 211 p.
3. Home safety education and provision of safety equipment for injury prevention / D. Kendrick et al. *Cochrane database of systematic reviews*. 2012. URL: <https://doi.org/10.1002/14651858.cd005014.pub3> (date of access: 29.06.2026).
4. Jullien S. Prevention of unintentional injuries in children under five years. *BMC pediatrics*. 2021. Vol. 21, S1. URL: <https://doi.org/10.1186/s12887-021-02517-2> (date of access: 26.06.2026).
5. Szeliski R. Computer vision. Cham : Springer International Publishing, 2022. URL: <https://doi.org/10.1007/978-3-030-34372-9> (date of access: 26.06.2026).
6. Dalal N., Triggs B. Histograms of oriented gradients for human detection. *2005 IEEE computer society conference on computer vision and pattern recognition (CVPR'05)*, San Diego, CA, USA. URL: <https://doi.org/10.1109/cvpr.2005.177> (date of access: 26.06.2026).
7. LeCun Y., Bengio Y., Hinton G. Deep learning. *Nature*. 2015. Vol. 521, no. 7553. P. 436–444. URL: <https://doi.org/10.1038/nature14539> (date of access: 26.06.2026).
8. Rich feature hierarchies for accurate object detection and semantic segmentation / R. Girshick et al. *2014 IEEE conference on computer vision and pattern recognition (CVPR)*, Columbus, OH, USA, 23–28 June 2014. 2014. URL: <https://doi.org/10.1109/cvpr.2014.81> (date of access: 26.06.2026).

9. Faster R-CNN: towards real-time object detection with region proposal networks / S. Ren et al. *IEEE transactions on pattern analysis and machine intelligence*. 2017. Vol. 39, no. 6. P. 1137–1149. URL: <https://doi.org/10.1109/tpami.2016.2577031> (date of access: 26.06.2026).
10. SSD: single shot multibox detector / W. Liu et al. *Computer vision – ECCV 2016*. Cham, 2016. P. 21–37. URL: https://doi.org/10.1007/978-3-319-46448-0_2 (date of access: 26.06.2026).
11. Focal loss for dense object detection / T.-Y. Lin et al. *2017 IEEE international conference on computer vision (ICCV)*, Venice, 22–29 October 2017. 2017. URL: <https://doi.org/10.1109/iccv.2017.324> (date of access: 26.06.2026).
12. You only look once: unified, real-time object detection / J. Redmon et al. *2016 IEEE conference on computer vision and pattern recognition (CVPR)*, Las Vegas, NV, USA, 27–30 June 2016. 2016. URL: <https://doi.org/10.1109/cvpr.2016.91> (date of access: 26.06.2026).
13. Object detection in 20 years: a survey / Z. Zou et al. *Proceedings of the IEEE*. 2023. P. 1–20. URL: <https://doi.org/10.1109/jproc.2023.3238524> (date of access: 26.06.2026).
14. AED-YOLO11: a small object detection model based on YOLO11 / X. Gong et al. *Digital signal processing*. 2025. Vol. 166. P. 105411. URL: <https://doi.org/10.1016/j.dsp.2025.105411> (date of access: 29.06.2026).
15. Meyers C., Löfstedt T., Elmroth E. Safety-critical computer vision: an empirical survey of adversarial evasion attacks and defenses on computer vision systems. *Artificial intelligence review*. 2023. URL: <https://doi.org/10.1007/s10462-023-10521-4> (date of access: 26.06.2026).
16. Microsoft COCO: common objects in context / T.-Y. Lin et al. *Computer vision – ECCV 2014*. Cham, 2014. P. 740–755. URL: https://doi.org/10.1007/978-3-319-10602-1_48 (date of access: 26.06.2026).

17. Vajgl M., Hurtik P., Nejezchleba T. Dist-YOLO: fast object detection with distance estimation. *Applied sciences*. 2022. Vol. 12, no. 3. P. 1354. URL: <https://doi.org/10.3390/app12031354> (date of access: 26.06.2026).

18. Absolute distance prediction based on deep learning object detection and monocular depth estimation models / A. Masoumian et al. *Frontiers in artificial intelligence and applications*. 2021. URL: <https://doi.org/10.3233/faia210151> (date of access: 26.06.2026).

19. Alumentations: fast and flexible image augmentations / A. Buslaev et al. *Information*. 2020. Vol. 11, no. 2. P. 125. URL: <https://doi.org/10.3390/info11020125> (date of access: 26.06.2026).

20. Li Y., Huang Y., Tao Q. Improving real-time object detection in Internet-of-Things smart city traffic with YOLOv8-DSAF method. *Scientific reports*. 2024. Vol. 14, no. 1. URL: <https://doi.org/10.1038/s41598-024-68115-1> (date of access: 26.06.2026).

21. Comprehensive analysis on Ultralytics-supported YOLO models for detection and recognition of large office objects for indoor navigation / R. G. Baldovino et al. *Procedia computer science*. 2024. Vol. 246. P. 3851–3858. URL: <https://doi.org/10.1016/j.procs.2024.09.158> (date of access: 29.06.2026).

22. Terven J., Córdova-Esparza D.-M., Romero-González J.-A. A comprehensive review of YOLO architectures in computer vision: from yolov1 to yolov8 and YOLO-NAS. *Machine learning and knowledge extraction*. 2023. Vol. 5, no. 4. P. 1680–1716. URL: <https://doi.org/10.3390/make5040083> (date of access: 28.06.2026).

23. Antiga L. P. G., Stevens E., Viehmann T. Deep learning with pytorch. Manning Publications Co. LLC, 2020.

24. Learning opencv 4 computer vision with python 3: get to grips with tools, techniques, and algorithms for computer vision and machine learning, 3rd edition. Packt Publishing, Limited, 2020. 372 p.

25. Khorasani M., Abdou M., Hernández Fernández J. Streamlit basics. *Streamlit for web development*. Berkeley, CA, 2025. P. 31–66. URL: https://doi.org/10.1007/979-8-8688-1826-4_2 (date of access: 29.06.2026).

26. A comparative analysis of object detection metrics with a companion open-source toolkit / R. Padilla et al. *Electronics*. 2021. Vol. 10, no. 3. P. 279. URL: <https://doi.org/10.3390/electronics10030279> (date of access: 26.06.2026).