

**НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ  
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ  
імені ІГОРЯ СІКОРСЬКОГО»**

**Факультет інформатики та обчислювальної техніки**

**Кафедра інформаційних систем та технологій**

До захисту допущено:

Завідувач кафедри

\_\_\_\_\_ Олександр РОЛІК

«\_\_» \_\_\_\_\_ 2025 р.

**Дипломний проєкт  
на здобуття ступеня бакалавра  
за освітньо-професійною програмою «Інформаційне забезпечення  
робототехнічних систем»  
спеціальності 126 «Інформаційні системи та технології»  
на тему: «Інтелектуальна роботизована система класифікації одягу»**

Виконав:

студент ІV курсу, групи ІК-13  
Савченко Володимир Євгенійович \_\_\_\_\_

Керівник:

ст. вик. кафедри ІСТ  
Моргаль Олег Михайлович \_\_\_\_\_

Рецензент:

доцент кафедри ІПІ, к.т.н., доцент.  
Лісовиченко Олег Іванович \_\_\_\_\_

Засвідчую, що у цьому дипломному проєкті немає запозичень з праць інших авторів без відповідних посилань.

Студент \_\_\_\_\_

Київ – 2025 року

**Національний технічний університет України**  
**«Київський політехнічний інститут імені Ігоря Сікорського»**  
**Факультет інформатики та обчислювальної техніки**  
**Кафедра інформаційних систем та технологій**

Рівень вищої освіти – перший (бакалаврський)

Спеціальність – 126 «Інформаційні системи та технології»

Освітньо-професійна програма «Інформаційне забезпечення робототехнічних систем»

ЗАТВЕРДЖУЮ

Завідувач кафедри

\_\_\_\_\_ Олександр РОЛІК

«\_\_» \_\_\_\_\_ 2025 р.

**ЗАВДАННЯ**

**на дипломний проєкт студенту**

**Савченку Володимирі Євгенійовичу**

1. Тема проєкту «Інтелектуальна роботизована система класифікації одягу», керівник проєкту ст. викладач Моргаль Олег Михайлович, затвердені наказом по університету від «23» травня 2025 р. № 1705-с
2. Термін подання студентом проєкту: 9 червня 2025 року
3. Вихідні дані до проєкту: відкритий датасет з Roboflow із зображеннями людей в одязі, інтелектуальна система класифікації одягу
4. Зміст пояснювальної записки: провести аналіз існуючих рішень та застосування класифікації одягу, методологія розробки системи класифікації одягу, розробити нейронну мережу, програмне забезпечення, описати робототехнічне забезпечення
5. Перелік графічного матеріалу (із зазначенням обов'язкових креслеників, плакатів, презентацій тощо): Блок схема (А3), Діаграма послідовності (А3), Діаграма потоку даних (А3), Діаграма розгортання (А3)
6. Дата видачі завдання 7 березня 2025 року

## Календарний план

№ з/п	Назва етапів виконання дипломного проєкту	Термін виконання етапів проєкту	Примітка
1.	Затвердження теми роботи	07.03.2025	
2.	Огляд літератури за тематикою	20.04.2025	
3.	Порівняльний аналіз існуючих рішень	24.04.2025	
4.	Формування цілей розробки	28.04.2025	
5.	Алгоритмізація поставленої задачі	08.05.2025	
6.	Аналіз та вибір можливих засобів для розробки системи	14.05.2025	
7.	Реалізація програмного забезпечення застосунку	28.05.2025	
8.	Аналіз та тестування продукту	30.05.2025	
9.	Оформлення пояснювальної записки	08.06.2025	

Студент

Володимир САВЧЕНКО

Керівник

Олег МОРГАЛЬ

## АНОТАЦІЯ

Савченко. В.Є. Інтелектуальна роботизована система класифікації одягу. КПІ ім. Ігоря Сікорського, Київ, 2025.

Проект містить 64 с. тексту, 24 рисунки, 5 таблиць, посилання на 20 літературних джерел, додатки та 4 конструкторських документів.

КЛАСИФІКАЦІЯ ОДЯГУ, КОМП'ЮТЕРНИЙ ЗІР, ЗГОРТКОВА НЕЙРОННА МЕРЕЖА, РЕАЛЬНИЙ ЧАС, ВЕБ-ЗАСТОСУНОК.

Об'єктом розробки є система автоматичної класифікації кількох елементів одягу на фото та відео з людьми.

Метою розробки є створення системи, здатної точно та ефективно розпізнавати типи одягу в умовах реального використання – на фотографіях з соціальних мереж, у вуличній моді чи маркетингових середовищах.

У дипломному проекті було реалізовано вебінтерфейс для завантаження та обробки зображень відео та з анотаціями результатів. Особливу увагу приділено обробці відео в реальному часі, розрахунку статистики присутності одягу за таймфреймами. Проект демонструє інтеграцію комп'ютерного зору з вебзастосунком, що відповідає актуальним тенденціям у розробці інтелектуальних робототехнічних систем.

Отримані результати забезпечують зручну взаємодію, а висока точність розпізнавання відкриває перспективи для застосування системи в електронній комерції, модній індустрії, системах безпеки та інших сферах, пов'язаних з аналізом зображень одягу.

## SUMMARY

Savchenko, V.E. Intelligent Robotic Clothing Classification System. National Technical University of Ukraine “Igor Sikorsky Kyiv Polytechnic Institute”, Kyiv, 2025.

The project contains 64 pages. text, 24 figures, 5 tables, links to 20 literary sources, annexes and 4 design documents.

Keywords: clothing classification, computer vision, convolutional neural network, real-time, web application.

The object of development is an automated system for classifying multiple clothing items in photos and videos featuring people.

The aim of the project is to create a system capable of accurately and efficiently recognizing types of clothing in real-world usage scenarios – including photos from social media, street fashion, or marketing environments.

This diploma project implements a web interface for uploading and processing images and videos with result annotations. Particular attention is given to real-time video processing and calculating statistics on clothing presence across time frames. The project demonstrates the integration of computer vision with a web application, aligning with current trends in the development of intelligent robotic systems.

The results obtained provide convenient interaction, and the high recognition accuracy opens up opportunities for applying the system in e-commerce, the fashion industry, security systems, and other areas related to the analysis of clothing images.

№ рядка	Формат	Позначення	Найменування	Кіл. аркушів	№ екз.	Примітка
1			<u>Документація загальна</u>			
2						
3			Знову розроблена			
4						
5	A4	ІК13.240БАК.006 ПЗ	Інтелектуальна роботизована	64		
6			система класифікації одягу.			
7			Пояснювальна записка			
8						
9	A3	ІК13.240БАК.006 Д1	Інтелектуальна роботизована	1		
10			система класифікації одягу.			
11			Блок схема			
12						
13	A3	ІК13.240БАК.006 Д2	Інтелектуальна роботизована	1		
14			система класифікації одягу.			
15			Діаграма послідовності			
16						
17	A3	ІК13.240БАК.006 Д3	Інтелектуальна роботизована	1		
18			система класифікації одягу.			
19			Діаграма потоку даних			
20						
21	A3	ІК13.240БАК.006 Д4	Інтелектуальна роботизована	1		
22			система класифікації одягу.			
23			Діаграма розгортки			
24						
25						
26						
27						
28						

				<b>ІК13.240БАК.006 ТП</b>						
Зм.	Лист	№ докум.	Підпис	Інтелектуальна роботизована система класифікації одягу. Відомість дипломного проекту			Літ.	Арк.	Аркушів	
Розробив	Савченко В.Є.						Т		1	1
Перевірив	Моргаль О.М.						КПІ ім. Ігоря Сікорського Група ІК-13			
Затв.										

**Пояснювальна записка**  
**до дипломного проєкту**  
**на тему: «Інтелектуальна роботизована система**  
**класифікації одягу»**

Київ – 2025 року

## ЗМІСТ

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ.....	5
ВСТУП.....	6
1 АНАЛІЗ ІСНУЮЧИХ РІШЕНЬ ТА ЗАСТОСУВАННЯ КЛАСИФІКАЦІЇ ОДЯГУ.....	8
1.1 Огляд проблеми класифікації одягу.....	8
1.2 Сфери застосування систем класифікації одягу .....	9
1.3 Технологічні основи систем класифікації одягу.....	10
1.4 Аналіз літератури з систем класифікації одягу.....	10
1.5 Практичні застосування систем класифікації одягу.....	11
1.5.1 Instagram.....	11
1.5.2 TikTok.....	12
1.5.3 Pinterest.....	13
Висновки до розділу 1 .....	14
2 МЕТОДОЛОГІЯ РОЗРОБКИ СИСТЕМИ КЛАСИФІКАЦІЇ ОДЯГУ .....	16
2.1 Вступ до технічних підходів класифікації кількох елементів одягу .....	16
2.2 Вибір мови програмування .....	17
2.2.1 Python.....	17
2.2.2 C++.....	18
2.2.3 Java.....	19
2.3 Вибір вебфреймворку .....	19
2.3.1 Django .....	20
2.3.2 Flask .....	21
2.3.3 FastAPI.....	22
2.4 Вибір моделі.....	23
2.4.1 Класичні CNN.....	23

				ІК13.240БАК.006 ПЗ					
Зм.	Лист	№ докум.	Підпис	Інтелектуальна роботизована система класифікації одягу. Пояснювальна записка		Літ.	Арк.	Аркушів	
Розробив	Савченко В.Є					Т		2	64
Перевірів	Моргаль О.М					КПІ ім. Ігоря Сікорського Група ІК-13			
Затв.									

2.4.2 Faster R-CNN.....	25
2.4.3 SSD.....	26
2.4.4 YOLOv11.....	26
2.5 Вибір бібліотек.....	28
2.5.1 Ultralytics.....	28
2.5.2 Roboflow.....	29
2.5.3 PyTorch.....	29
2.5.4 OpenCV.....	30
Висновок до розділу 2.....	30
<b>3 РОЗРОБЛЕННЯ НЕЙРОННОЇ МЕРЕЖІ.....</b>	<b>32</b>
3.1 Підготовка даних для тренування моделі.....	32
3.1.1 Вибір і структура датасету.....	32
3.1.2 Анотація через Roboflow.....	34
3.1.3 Формування та експорт датасету через Roboflow.....	35
3.2 Конфігурація моделі.....	36
3.3 Тренування нейронної мережі.....	39
3.4 Оцінка ефективності моделі.....	40
3.5 Тестування моделі.....	43
3.5.1 Швидкість роботи.....	43
3.5.2 Аналіз складних випадків.....	44
Висновки до розділу 3.....	46
<b>4 РОЗРОБЛЕННЯ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ.....</b>	<b>48</b>
4.1 Архітектура системи.....	48
4.2 Основні компоненти системи.....	48
4.3 Взаємодія між компонентами.....	50
4.4 Особливості архітектури та переваги.....	53
Висновки до розділу 4.....	54
<b>5 РОБОТОТЕХНІЧНЕ ЗАБЕЗПЕЧЕННЯ.....</b>	<b>56</b>
5.1 Інтеграція з робототехнічними системами.....	56
5.2 Апаратна частина робототехнічного забезпечення.....	58

5.3 Передача та обробка даних .....	59
Висновки до розділу 5 .....	60
ВИСНОВКИ.....	61
ПЕРЕЛІК ВИКОРИСТАНИХ ДЖЕРЕЛ.....	63
ДОДАТОК А.....	65
ДОДАТОК Б .....	66

					ІК13.240БАК.006 ПЗ	Арк.
Зм.	Лист	№ докум.	Підпис	Дата		4

## ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ

Датасет – набір даних, який використовується для навчання та тестування моделей комп'ютерного зору.

Ендпоінт – точка контакту між клієнтом API і сервером API.

Шопінг – процес покупки різноманітних товарів.

ASGI – Asynchronous Server Gateway Interface.

API – Application Programming Interface.

JSON – JavaScript Object Notation.

FPS – Frames Per Second.

					ІК13.240БАК.006 ПЗ	Арк.
Зм.	Лист	№ докум.	Підпис	Дата		5

## ВСТУП

Щодня мільйони людей публікують фотографії в соціальних мережах, створюють візуальний контент для маркетингових кампаній або знімають відео зі свого повсякденного життя. Цей потік даних формує унікальне середовище, де зображення одягу стають цінним джерелом інформації для аналізу модних трендів, поведінки споживачів і навіть безпеки. Разом із цим зростає потреба в інтелектуальних системах, здатних автоматично класифікувати одяг на зображеннях або відеопотоці в реальному часі. Така автоматизація дозволяє значно скоротити час і ресурси, необхідні для обробки великих обсягів візуального контенту, що є особливо актуальним у сучасних умовах цифрової економіки.

Попит на подібні технології зумовлений кількома ключовими факторами. По-перше, стрімкий розвиток електронної комерції вимагає інструментів для автоматичного тегування товарів на зображеннях, що полегшує пошук і персоналізовані рекомендації для користувачів. Наприклад, платформи для шопінгу уже застосовують комп'ютерний зір для ідентифікації одягу, але їхні системи часто мають обмежену точність у складних сценаріях, таких як вулична мода чи перекриття об'єктів [1]. По-друге, аналіз модних трендів у соціальних мережах, дозволяє брендам прогнозувати попит і адаптувати асортимент, що є критично важливим у швидкоплинній індустрії моди [2]. По-третє, системи класифікації одягу знаходять застосування в безпеці, зокрема для ідентифікації осіб за одягом на камерах спостереження, а також в аналітиці споживчої поведінки [3]. Ручна класифікація зображень є трудомістким і неефективним процесом, що підкреслює потребу в автоматизованих рішеннях, які поєднують високу точність, швидкість і адаптивність до реальних умов.

Основні проєктні рішення полягають у використанні моделі YOLOv11 через її одноступеневий підхід, що дозволяє одночасно локалізувати та класифікувати кілька елементів одягу, таких як футболки, штани чи взуття, навіть у складних сценаріях. Для розробки користувацького інтерфейсу використано вебфреймворк

					ІК13.240БАК.006 ПЗ	Арк.
						6
Зм.	Лист	№ докум.	Підпис	Дата		

Django. Бібліотека OpenCV застосовується для попередньої обробки зображень, забезпечуючи нормалізацію та стабільність роботи моделі. PyTorch обрано для реалізації YOLOv11 завдяки своїй гнучкості та динамічному обчислювальному графу, що полегшує налаштування та експерименти з архітектурою моделі. Крім того, було створено власний набір даних із ручною анотацією, який адаптовано до реальних сценаріїв використання, що підвищує практичну цінність системи.

Можливі галузі застосування результатів проекту охоплюють електронну комерцію – для автоматичного тегування товарів в інтернет-магазинах, полегшуючи пошук і рекомендації; маркетинг – для аналізу модних трендів у соціальних мережах, прогнозування попиту та персоналізованих рекламних кампаній; а також сферу безпеки – для ідентифікації осіб за одягом на камерах спостереження.

Дипломний проект складається зі вступу, п'яти основних розділів, висновків, списку використаних джерел із 20 найменувань, додатків та графічної частини, що включає 4 кресленики формату А3. Загальний обсяг текстової частини становить 64 сторінки.

					ІК13.240БАК.006 ПЗ	Арк.
Зм.	Лист	№ докум.	Підпис	Дата		7

# 1 АНАЛІЗ ІСНУЮЧИХ РІШЕНЬ ТА ЗАСТОСУВАННЯ КЛАСИФІКАЦІЇ ОДЯГУ

## 1.1 Огляд проблеми класифікації одягу

Автоматична класифікація одягу за типом (футболка, штани, сукня, куртка тощо) на основі фотографій із людьми є складним завданням, яке вимагає високої точності розпізнавання в умовах реального світу. Це завдання потребує високої точності розпізнавання в умовах реального світу, де одяг на людях відрізняється за стилем, кольором, текстурою та позою. Додаткові труднощі створюють складні фони – наприклад, міське середовище чи інтер'єри – які додають шум до зображень і ускладнюють ідентифікацію елементів одягу.

На рисунку 1.1 можна побачити, що одяг іноді частково закритий аксесуарами, об'єктами чи іншими людьми, що ще більше знижує ефективність класифікації. Окрім цього, швидка зміна модних трендів вимагає від моделей адаптивності до нових стилів, які ще не представлені у тренувальних наборах даних.



Рисунок 1.1 – Приклад проблем класифікації

Ще одним викликом при створенні системи класифікації одягу є потреба у великих та різноманітних наборах даних для якісного навчання моделей. Сучасні

					ІК13.240БАК.006 ПЗ	Арк.
Зм.	Лист	№ докум.	Підпис	Дата		8

фотографії, на яких зображено людей, часто містять одяг у різних ракурсах, з різними фонами та в умовах освітлення, які можуть змінюватися залежно від контексту зйомки. Це створює додаткові труднощі для розпізнавання одягу, оскільки модель має навчитися правильно виділяти та класифікувати елементи навіть у складних сценаріях. Наприклад, датасет DeepFashion2 є цінним ресурсом завдяки своїй різноманітності (~800 000 зображень) та анотаціям, що допомагають якісно розпізнавати одяг у складних сценах [4]. Швидкість обробки зображень також критична для інтерактивних застосунків, де користувачі очікують результатів у режимі реального часу.

Незважаючи на ці труднощі, системи класифікації одягу мають значний потенціал для аналізу модних тенденцій і підтримки маркетингових стратегій. Подолання викликів потребує використання передових методів комп'ютерного зору та створення спеціалізованих наборів даних, що відображають реальні сценарії.

## 1.2 Сфери застосування систем класифікації одягу

Інтелектуальні системи класифікації одягу є важливими для маркетингових застосувань, зокрема для аналізу модних трендів і поведінки споживачів у соціальних мережах. Такі системи дозволяють брендам і ритейлерам автоматично визначати популярні типи одягу, стилі та кольори на основі фотографій, опублікованих у мережах, таких як Instagram чи TikTok, що допомагає прогнозувати ринковий попит. Наприклад, виявлення зростання популярності широких джинсів чи кроп-топів може вплинути на асортиментну стратегію компаній.

У сфері електронної комерції класифікація одягу на фотографіях із людьми підтримує рекомендаційні системи, пропонуючи користувачам товари, схожі на ті, що вони бачили в соціальних мережах. Це підвищує залученість клієнтів і сприяє зростанню продажів. Крім того, аналіз одягу на людях дозволяє створювати персоналізовані рекламні кампанії, орієнтовані на конкретні демографічні групи чи стилістичні вподобання.

					ІК13.240БАК.006 ПЗ	Арк.
						9
Зм.	Лист	№ докум.	Підпис	Дата		

Системи класифікації також знаходять застосування в інших галузях. Наприклад, у fashion-аналітиці вони допомагають відстежувати еволюцію стилів, аналізуючи фотографії з модних показів чи вуличної моди [2]. У криміналістиці класифікація одягу на зображеннях із камер спостереження може сприяти ідентифікації осіб [3]. Таким чином, ці системи є універсальними інструментами для аналізу візуальних даних, що мають значення для маркетингу, ритейлу та інших сфер.

### 1.3 Технологічні основи систем класифікації одягу

Системи класифікації одягу базуються на комп'ютерному зорі та методах штучного інтелекту, які дозволяють розпізнавати типи одягу на основі їхніх візуальних характеристик, таких як форма, текстура чи колір. У сценаріях із людьми, наприклад, на фотографіях із соціальних мереж, алгоритми повинні враховувати пози, ракурси та часткове перекриття одягу, що ускладнює завдання. Глибокі нейронні мережі, зокрема згорткові (CNN), є основним інструментом для таких систем, оскільки вони здатні автоматично виділяти ключові ознаки зображень. Наприклад, архітектури, такі як Inception-v3, оптимізують обробку великих обсягів даних, що важливо для аналізу трендів у реальному часі [5].

Для підвищення точності класифікації часто застосовується трансферне навчання, яке дозволяє адаптувати попередньо навчені моделі до специфічних завдань, таких як розпізнавання одягу на людях. Однак ефективність таких систем залежить від якості даних, які мають включати різноманітні приклади одягу в реальних умовах. Попередня обробка зображень, як-от сегментація чи видалення фону, також відіграє важливу роль у підвищенні точності.

### 1.4 Аналіз літератури з систем класифікації одягу

Дослідження класифікації одягу охоплюють широкий спектр методів і застосувань. Традиційні підходи, такі як використання гістограм орієнтованих градієнтів (HOG), забезпечували базову точність для простих зображень, але були

					ІК13.240БАК.006 ПЗ	Арк.
						10
Зм.	Лист	№ докум.	Підпис	Дата		

менш ефективними для фотографій із людьми через складні фони та пози [6]. Сучасні методи глибокого навчання значно покращили результати. Наприклад, Liu et al. (2016) розробили модель FashionNet, яка досягає точності 66% для класифікації одягу на зображеннях із соціальних мереж [7]. Yamaguchi et al. (2014) запропонували систему для аналізу вуличної моди, що визначає популярні стилі з точністю 78% [8].

У технічному плані Szegedy et al. (2016) вдосконалили архітектуру Inception, досягнувши точності 85% для обробки великих обсягів даних у реальному часі [5]. Nara et al. (2016) застосували CNN для аналізу одягу на модних показах, що сприяє прогнозуванню трендів із точністю 80% [2]. Ці дослідження підкреслюють універсальність і потенціал систем класифікації одягу.

### 1.5 Практичні застосування систем класифікації одягу

Штучний інтелект, зокрема комп'ютерний зір, відіграє ключову роль у вдосконаленні сервісів онлайн-комерції, забезпечуючи персоналізацію, аналіз трендів і автоматизацію обробки зображень. Моделі комп'ютерного зору дозволяють розпізнавати кілька елементів одягу на одному фото, що сприяє ефективному тегуванню, класифікації та рекомендаціям. Розглянемо застосування таких технологій на прикладі Instagram, TikTok і Pinterest.

#### 1.5.1 Instagram



Рисунок 1.2 – Домашня сторінка Instagram

Instagram – провідна соціальна мережа, що спеціалізується на візуальному контенті, включаючи моду та стиль. З його інтерфейсом можна ознайомитись на рисунку 1.2. Платформа дозволяє користувачам, брендам і впливовим особам публікувати фото та відео, які часто демонструють одяг і аксесуари. Instagram Shopping інтегрує функції комерції, дозволяючи тегувати товари безпосередньо на зображеннях, що спрощує покупки.

Комп’ютерний зір на Instagram використовується для автоматичного тегування одягу на фото та відео. Алгоритми аналізують зображення, розпізнаючи кілька елементів одягу (наприклад, сукня, сумка, взуття) і додаючи теги, такі як категорія (сукня), стиль (бохо), колір (бежевий) або бренд. Ці теги генеруються за допомогою нейронних мереж, які сегментують об’єкти та класифікують їх за атрибутами. Наприклад, пост із фото блогера в оверсайз-светрі та джинсах може отримати теги “светр”, “джинси”, “кежуал”, які відображаються в Instagram Shopping для переходу до покупки. Аналогічна технологія розпізнавання об’єктів застосовується в Automatic alt text, де створюються текстові описи для людей із вадами зору, наприклад, “Людина в синьому светрі та джинсах” [9].

### 1.5.2 TikTok

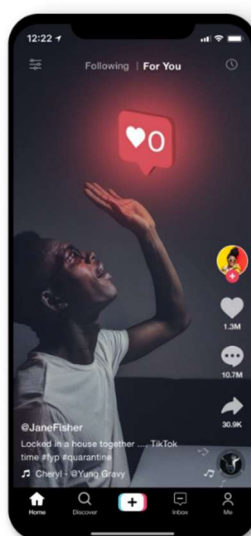


Рисунок 1.3 – Домашня сторінка TikTok

					ІК13.240БАК.006 ПЗ	Арк.
Зм.	Лист	№ докум.	Підпис	Дата		12

TikTok – платформа коротких відео, популярна серед молоді, яка активно впливає на модні тренди завдяки вірусному контенту. Користувачі створюють відео з демонстрацією одягу, стилізацій і DIY-проектів, що сприяє популяризації швидкої моди та локальних брендів. Домашню сторінку TikTok показано на рисунку 1.3.

Комп’ютерний зір на TikTok застосовується для тегування та опису одягу у відео. Моделі розпізнають кілька елементів одягу в кадрі (наприклад, кроп-топ, широкі штани, кеди) і автоматично додають теги за категоріями, кольорами чи стилями (наприклад, “кроп-топ”, “білий”, “стритстайл”). Ці теги використовуються для інтеграції з комерційними функціями, такими як TikTok Shop, де користувачі можуть купувати товари, показані у відео [10]. Наприклад, відео з танцем у мінімалістичній сукні може отримати теги “сукня”, “чорна”, “вечірній стиль”, що полегшує пошук схожих товарів. Алгоритми також аналізують вірусні тренди, допомагаючи брендам адаптувати асортимент до попиту на певні стилі, наприклад, на оверсайз-куртки.

### 1.5.3 Pinterest

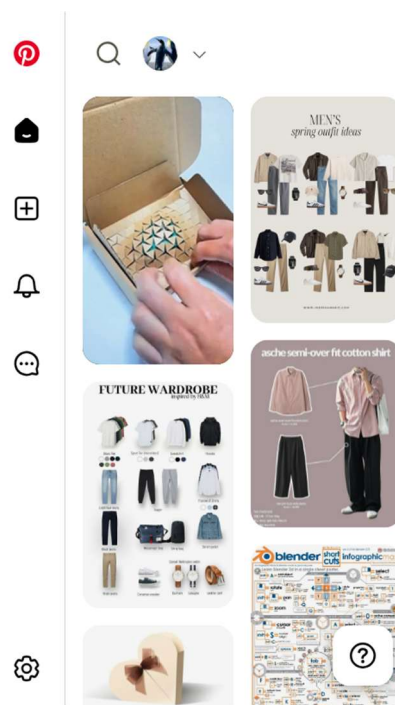


Рисунок 1.4 – Домашня сторінка Pinterest

					ІК13.240БАК.006 ПЗ	Арк.
Зм.	Лист	№ докум.	Підпис	Дата		13

Pinterest – це соціальна платформа, орієнтована на візуальний контент, яка дозволяє користувачам знаходити натхнення для моди, стилю, дизайну та інших сфер життя через збереження та організацію зображень (Pin-ів). Користувачі, бренди та інфлюенсери публікують фотографії одягу, аксесуарів і стилізацій, як це зображено на рисунку 1.4, що робить Pinterest важливим інструментом для аналізу модних трендів і споживчих уподобань.

Комп'ютерний зір на Pinterest відіграє ключову роль у розпізнаванні та класифікації одягу на зображеннях, використовуючи сучасні методи машинного навчання та штучного інтелекту. Алгоритми платформи аналізують Pin-и, ідентифікуючи кілька елементів одягу в кадрі (наприклад, блуза, спідниця, сумка) і додаючи теги за категоріями, стилями чи кольорами (наприклад, «блуза», «квітковий принт», «білий»). Ці теги генеруються за допомогою нейронних мереж, які сегментують об'єкти та класифікують їх за атрибутами [11]. Наприклад, зображення блогера в джинсовій спідниці та кроп-топі може отримати теги «спідниця», «джинсова», «кежуал», що відображаються в рекомендаціях або комерційних функціях платформи.

Таким чином, комп'ютерний зір на Pinterest підтримує автоматизацію тегування, персоналізацію рекомендацій і аналіз модних трендів, сприяючи розвитку електронної комерції та модної індустрії.

## Висновки до розділу 1

Аналіз систем класифікації одягу підтверджує їхню роль як потужного інструменту для маркетингу, fashion-аналітики, електронної комерції, криміналістики та культурних досліджень. Основні виклики включають варіативність одягу на фотографіях із людьми, складні фони та потребу в різноманітних наборах даних для реальних сценаріїв, таких як вулична мода чи соціальні мережі. Сучасні методи глибокого навчання, зокрема згорткові нейронні мережі (CNN) і трансферне навчання, дозволяють досягати точності до 85% у складних умовах [5, 6].

					ІК13.240БАК.006 ПЗ	Арк.
Зм.	Лист	№ докум.	Підпис	Дата		14

Класифікація одягу сприяє прогнозуванню модних трендів, персоналізації маркетингових стратегій і підтримці рекомендаційних систем, що підвищує продажі та розуміння споживчих уподобань. Застосування в криміналістиці, віртуальних примірочних і аналізі культурної спадщини підкреслює універсальність технологій [7, 12]. Подолання обмежень, таких як недостатність даних і швидкість обробки, відкриває перспективи для створення гнучких систем, що адаптуються до динамічних ринкових і соціальних потреб.

Використання великих і різноманітних наборів даних, таких як DeepFashion2, у поєднанні з хмарними технологіями, відкриває можливості для масштабування систем і обробки даних у реальному часі.

					ІК13.240БАК.006 ПЗ	Арк.
Зм.	Лист	№ докум.	Підпис	Дата		15

## 2 МЕТОДОЛОГІЯ РОЗРОБКИ СИСТЕМИ КЛАСИФІКАЦІЇ ОДЯГУ

### 2.1 Вступ до технічних підходів класифікації кількох елементів одягу

Аналіз літератури та викликів, представлений у першому розділі, підкреслив складність автоматичної класифікації одягу, особливо в умовах реальних фотографій із людьми, де на одному зображенні можуть бути присутні кілька елементів одягу. Розпізнавання кількох об'єктів на одному фото є актуальним завданням, оскільки воно відображає типові сценарії соціальних мереж, вуличної моди чи маркетингових застосувань, де необхідно ідентифікувати повний образ, а не окремі предмети. Цей розділ присвячено методології розробки інтелектуальної системи, здатної класифікувати та локалізувати кілька елементів одягу на одному зображенні, з акцентом на сучасні алгоритми комп'ютерного зору та глибокого навчання.

Класифікація кількох елементів одягу на одному фото ускладнюється через перекриття об'єктів, варіативність стилів і текстур, а також вплив фону, що вимагає використання методів детекції об'єктів і сегментації [1]. Наприклад, на фотографії людини в міському середовищі сорочка може частково перекриватися курткою, а штани – аксесуарами, що знижує точність розпізнавання. Для вирішення цих проблем застосовуються архітектури, такі як YOLO чи Faster R-CNN, які дозволяють одночасно локалізувати та класифікувати кілька об'єктів на зображенні [7]. Крім того, створення спеціалізованих наборів даних із анотаціями для кількох елементів одягу є критично важливим для навчання таких моделей.

У цьому розділі розглядаються основні етапи розробки системи: вибір архітектури моделі, методи попередньої обробки зображень, створення та анотація наборів даних, а також оцінка ефективності системи в реальних умовах. Особлива увага приділяється адаптації алгоритмів до маркетингових застосувань, таких як аналіз трендів у соціальних мережах, де розпізнавання повних образів може сприяти персоналізації рекомендацій і прогнозуванню попиту.



перше місце з часткою 25.35%, що підтверджує його популярність серед розробників і дослідників. Простота синтаксису робить Python легким для швидкого прототипування, що є критично важливим для проєктів із обмеженим часом розробки.

Python має розвинену екосистему бібліотек, таких як PyTorch, OpenCV, NumPy і Scikit-learn, які ідеально підходять для задач класифікації одягу.

Велика та активна спільнота Python забезпечує доступ до документації та рішень технічних проблем, що полегшує розробку. Python також є стандартом для машинного навчання та аналізу даних, що відповідає маркетинговим застосуванням проєкту, таким як обробка зображень із соціальних мереж для прогнозування трендів. Таким чином, Python є оптимальним вибором завдяки поєднанню простоти, бібліотек і підтримки спільноти.

### 2.2.2 C++

C++ є мовою програмування, відомою своєю високою продуктивністю та низькорівневим контролем над апаратними ресурсами, що робить її цінною для застосувань, де швидкість виконання є пріоритетом. З часткою 9.94% у TIOBE Index, C++ займає значну нішу, зокрема в розробці ігрових двигунів і високопродуктивних систем. У контексті комп'ютерного зору C++ часто використовується для створення фреймворків глибокого навчання, таких як частини TensorFlow або OpenCV, завдяки здатності оптимізувати обчислення.

Для проєкту класифікації одягу C++ міг би бути корисним для прискорення обробки зображень або передбачення моделі на апаратному рівні. Однак C++ має складний синтаксис і вимагає ручного керування пам'яттю, що уповільнює розробку порівняно з Python. Відсутність такої багатої екосистеми бібліотек для машинного навчання, як у Python, ускладнює швидке тренування моделей і інтеграцію з вебфреймворками. Крім того, менш активна підтримка спільноти для специфічних задач комп'ютерного зору робить C++ менш зручним для цього

					ІК13.240БАК.006 ПЗ	Арк.
Зм.	Лист	№ докум.	Підпис	Дата		18

проєкту. Таким чином, попри високу продуктивність, C++ поступається Python за швидкістю розробки та гнучкістю.

### 2.2.3 Java

Java – це об’єктно-орієнтована мова програмування з часткою 9.31% у TIOBE Index, яка вирізняється своєю платформонезалежністю та зрілою екосистемою, що робить її популярною в корпоративних середовищах. Java підходить для розробки масштабних систем штучного інтелекту, таких як рекомендаційні платформи в електронній комерції, завдяки бібліотекам, таким як DeepLearning4j і Weka. Її сильні сторони включають надійність, масштабованість і підтримку великих проєктів із розподіленими системами.

У контексті цього проєкту Java могла б бути використана для створення серверної частини вебзастосунку, альтернативної Django, або для інтеграції моделей комп’ютерного зору в корпоративні системи. Однак Java має вищу складність порівняно з Python для задач швидкого прототипування, а її бібліотеки для комп’ютерного зору (наприклад, OpenCV-Java) менш розвинені, ніж аналоги в Python [14]. Крім того, тренування моделей у Java є менш зручним через обмежену підтримку фреймворків глибокого навчання порівняно з PyTorch. Для маркетингових застосувань, де потрібна швидка обробка зображень із соціальних мереж і гнучка інтеграція з вебінтерфейсом, Java поступається Python за простотою та швидкістю розробки.

### 2.3 Вибір вебфреймворку

Вебфреймворк – це набір інструментів і бібліотек, які спрощують створення вебзастосунків, надаючи розробнику готову структуру для обробки запитів користувачів, відображення інтерфейсу та інтеграції з базами даних або зовнішніми сервісами. Для цього проєкту, що передбачає класифікацію одягу на фото та відео з подальшим відображенням результатів у зручному для користувача інтерфейсі,

					ІК13.240БАК.006 ПЗ	Арк.
Зм.	Лист	№ докум.	Підпис	Дата		19

вебфреймворк дає змогу швидко реалізувати завантаження зображень та відео, передачу їх на обробку комп'ютерним зором і виведення анотованих результатів. Вибір вебфреймворку є критичним для забезпечення зручного користувацького інтерфейсу, обробки великих файлів, а також інтеграції з алгоритмами комп'ютерного зору.

### 2.3.1 Django

Django – це високорівневий вебфреймворк для Python, який здобув популярність завдяки своїй повноцінній архітектурі та філософії «батарейки включено», що передбачає наявність усього необхідного для створення безпечних і масштабованих вебзастосунків прямо з коробки. Розроблений для швидкого створення структурованих проєктів, Django спирається на архітектуру MVT (Model-View-Template), де моделі відповідають за дані, представлення обробляють логіку запитів, а шаблони формують інтерфейс. Ця архітектура показана на Рисунку 2.2. Такий підхід забезпечує чітке розділення обов'язків, дозволяючи розробникам зосередитися на функціональності, а не на базовій конфігурації.. Для обробки файлів фреймворк пропонує надійні механізми, що дозволяють безпечно завантажувати та зберігати великі файли, такі як відео до 100 МБ, що є критично важливим для цього проєкту.

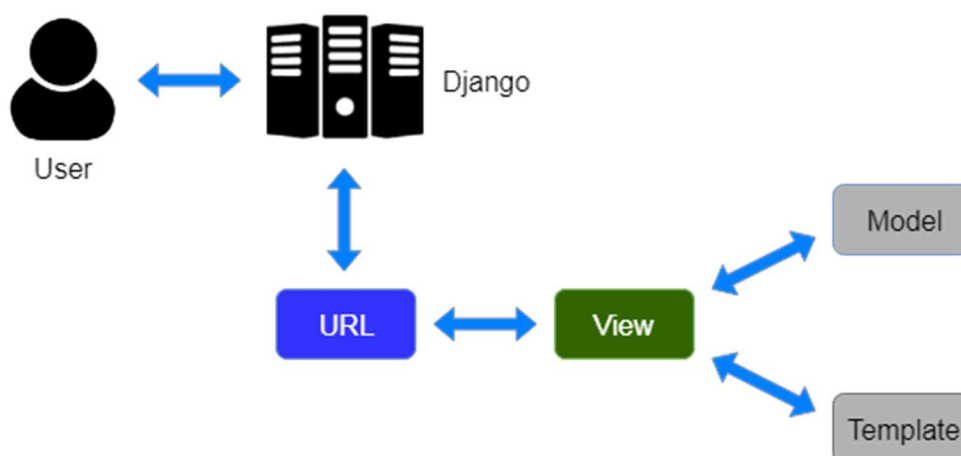


Рисунок 2.2 – Архітектура MVT

Екосистема Django вражає своєю широтою: сторонні пакети, такі як `django-rest-framework` для API чи `django-storages` для хмарного зберігання, розширюють можливості фреймворку. Масштабованість забезпечується підтримкою розгортання на серверах через `Gunicorn` і `Nginx`, що робить Django придатним для застосунків із високим навантаженням. Активна спільнота та детальна документація полегшують вирішення технічних питань. У контексті цього проєкту Django ідеально підходить для створення вебінтерфейсу, який дозволяє користувачам завантажувати фото чи відео, передавати їх до Python-скриптів для передбачень і відображати результати, такі як теги одягу. Django, наприклад, використовується в Instagram, що підтверджує його здатність справлятися з подібними сценаріями [9]. Таким чином, Django вирізняється як надійний і зручний фреймворк для цього проєкту.

### 2.3.2 Flask

Flask – це мікрофреймворк для Python, який завоював популярність завдяки своїй простоті та гнучкості, пропонуючи мінімалістичний підхід до веброзробки. На відміну від повноцінних фреймворків, Flask дотримується філософії «зроби сам», надаючи лише базові компоненти, такі як маршрутизація URL і обробка HTTP-запитів, і залишаючи решту на розсуд розробника. Така архітектура дозволяє створювати легкі застосунки з мінімальним споживанням ресурсів, що робить Flask ідеальним для швидких прототипів або невеликих проєктів. Замість вбудованих інструментів, як у Django, Flask покладається на сторонні бібліотеки: `Flask-SQLAlchemy` забезпечує роботу з базами даних, `Flask-WTF` обробляє форми, а `Flask-Login` керує автентифікацією. Ця модульність дає розробникам свободу вибору, дозволяючи налаштувати застосунок під конкретні потреби, але водночас вимагає додаткових зусиль для конфігурації. Наприклад, для обробки великих файлів, таких як відео до 100 МБ, потрібні плагіни на кшталт `Flask-Uploads`.

Екосистема Flask є багатогою, але менш інтегрованою порівняно з Django, що може ускладнити розробку складних систем. Його легкий синтаксис і низька крива

					ІК13.240БАК.006 ПЗ	Арк.
Зм.	Лист	№ докум.	Підпис	Дата		21

навчання роблять фреймворк привабливим для новачків, а компактність коду сприяє швидкому створенню API чи базових вебінтерфейсів. У контексті цього проєкту Flask міг би бути використаний для створення простого API, що передає зображення до моделі і повертає теги одягу. Однак необхідність ручного налаштування для обробки великих файлів, стабільного вебінтерфейсу робить Flask менш зручним порівняно з Django. Для маркетингових задач, де потрібна надійна структура та швидке розроблення, Flask вимагає більше часу на інтеграцію модулів, що знижує його ефективність. Таким чином, попри гнучкість, Flask поступається Django за функціональністю та зручністю для цього проєкту.

### 2.3.3 FastAPI

FastAPI – це сучасний фреймворк для Python, який революціонує розробку API завдяки своїй асинхронній архітектурі та високій продуктивності [19]. Побудований на базі Starlette і Python 3.6+ (з використанням `asyncio`), FastAPI використовує синтаксис `async/await` для обробки тисяч запитів одночасно, що робить його одним із найшвидших фреймворків у бенчмарках – до 300% швидше за Django в тестах API [20]. Його унікальність полягає в інтеграції типізації через `Pydantic`, що забезпечує строгую валідацію вхідних даних і зменшує кількість помилок. FastAPI автоматично генерує документацію API (на основі `OpenAPI/Swagger`), дозволяючи розробникам і тестувальникам швидко перевірити ендпоінти без додаткових інструментів. Такий підхід робить фреймворк ідеальним для створення високонавантажених систем, таких як мікросервіси чи API для реального часу.

Екосистема FastAPI підтримує інтеграцію з базами даних через `SQLAlchemy` чи `Tortoise ORM`, а також із хмарними сервісами, але вона менш розвинена для створення повноцінних вебінтерфейсів із шаблонами. FastAPI фокусується на API, що обмежує його можливості для проєктів, де потрібен графічний інтерфейс. Для реалізації вебінтерфейсу необхідно використовувати додаткові бібліотеки, такі як `Jinja2`, або поєднувати FastAPI з фронтенд-фреймворками, що додає складності.

					ІК13.240БАК.006 ПЗ	Арк.
Зм.	Лист	№ докум.	Підпис	Дата		22

Крива навчання FastAPI є вищою порівняно з Django чи Flask через асинхронне програмування та типізацію. У контексті цього проєкту FastAPI міг би ефективно обробляти API-запити для передачі зображень до моделі, особливо в сценаріях із масовим аналізом фото з соціальних мереж. Однак створення вебінтерфейсу для завантаження фото/відео та відображення результатів потребує додаткових зусиль, а обробка великих файлів, хоч і можлива, не є настільки інтегрованою, як у Django. Таким чином, FastAPI поступається за зручністю для проєкту з акцентом на вебінтерфейс.

## 2.4 Вибір моделі

У системах комп'ютерного зору під моделлю машинного навчання розуміють алгоритм або програмний блок, який навчається на основі великої кількості даних для виконання певних завдань, таких як класифікація або детекція об'єктів на зображеннях. У контексті задачі класифікації одягу модель повинна не лише розпізнавати категорію об'єкта (наприклад, светр, джинси, сукня), але й точно визначати його розташування на зображенні – навіть якщо одяг частково перекритий аксесуарами чи іншими об'єктами.

Для таких завдань найефективнішими є згорткові нейронні мережі (CNN), які дозволяють автоматично витягувати ієрархічні ознаки зображень, починаючи від базових текстур і контурів до більш складних патернів та форм. CNN мають перевагу перед класичними методами комп'ютерного зору (що базуються на ручному визначенні ознак), оскільки вони здатні ефективно навчатися на великих об'ємах даних і працювати у складних умовах – наприклад, зі змінним освітленням, різними ракурсами або фоном.

### 2.4.1 Класичні CNN

Класичні згорткові нейронні мережі, такі як VGG16 і ResNet, є фундаментальними моделями для обробки зображень, що здобули визнання

					ІК13.240БАК.006 ПЗ	Арк.
Зм.	Лист	№ докум.	Підпис	Дата		23





реалізації, пов'язана з налаштуванням двох мереж (RPN і класифікатора), також збільшує час розробки. Таким чином, попри високу точність, Faster R-CNN поступається за швидкістю та практичністю, що обмежує її застосування у системах, де критичними є низька затримка та спрощене розгортання. Ця модель більше підходить для задач, що не вимагають обробки в реальному часі та де пріоритетом є максимальна точність.

### 2.4.3 SSD

SSD (Single Shot MultiBox Detector) – це одноступенева модель, яка виконує локалізацію та класифікацію об'єктів за один прохід нейронної мережі, забезпечуючи вищу швидкість порівняно з двоступеневими моделями. На наборі даних PASCAL VOC 2007 модель SSD512 досягає mAP 76.8% при швидкості 19 кадрів за секунду (batch size 1), що робить її ефективною для швидкої обробки великих об'єктів, таких як куртки чи штани на фотографіях із соціальних мереж [18]. Архітектура SSD використовує згорткові шари з різними масштабами для детекції об'єктів різного розміру, а простота реалізації сприяє швидкому розгортанню. Однак точність SSD знижується при розпізнаванні малих об'єктів, таких як взуття чи топ, які часто присутні в образах. У сценаріях із перекриттям одягу чи складними фонами модель може пропускати об'єкти.

### 2.4.4 YOLOv11

YOLOv11, остання ітерація серії Ultralytics YOLO, є передовою моделлю для детекції об'єктів у реальному часі, яка поєднує високу точність і швидкість, ідеально відповідаючи задачам класифікації одягу.. Її архітектура включає три ключові компоненти: backbone, neck і head, які забезпечують ефективне витягнення ознак і передбачення. Backbone використовує C3k2, який замінює C2f блок YOLOv8, застосовуючи два малі конволюції замість однієї великої, що зменшує обчислювальну складність і прискорює обробку. Spatial Pyramid Pooling-Fast

					ІК13.240БАК.006 ПЗ	Арк.
						26
Зм.	Лист	№ докум.	Підпис	Дата		

(SPPF) оптимізує пулінг ознак для різних масштабів, дозволяючи точно розпізнавати великі (светри, сукні) і малі (взуття) об'єкти. Neck інтегрує C3k2 і C2PSA (Cross Stage Partial with Spatial Attention), який фокусується на ключових регіонах зображення, покращуючи детекцію. Head використовує C3k2, CBS (Convolution-BatchNorm-SiLU) шари і Detect layer для генерації обмежувальних рамок і класів, забезпечуючи швидке передбачення за 0.016 секунди на зображення [12].

Одноступеневий підхід YOLOv11 виконує детекцію і класифікацію за один прохід, використовуючи на 16% менше параметрів (~9.4 млн для YOLOv11s) порівняно з YOLOv8s (~11.2 млн), що сприяє ефективному розгортанню на edge-пристроях і хмарних платформах. Модель підтримує багатозадачність, включаючи детекцію, сегментацію, класифікацію, оцінку пози та орієнтовану детекцію (OBB), що дозволяє не лише локалізувати одяг, але й аналізувати його форму чи орієнтацію, наприклад, для образів у соціальних мережах. YOLOv11 доступна у варіантах від nano (для мобільних пристроїв) до xlarge (для високопродуктивних серверів), що забезпечує гнучкість для різних сценаріїв.

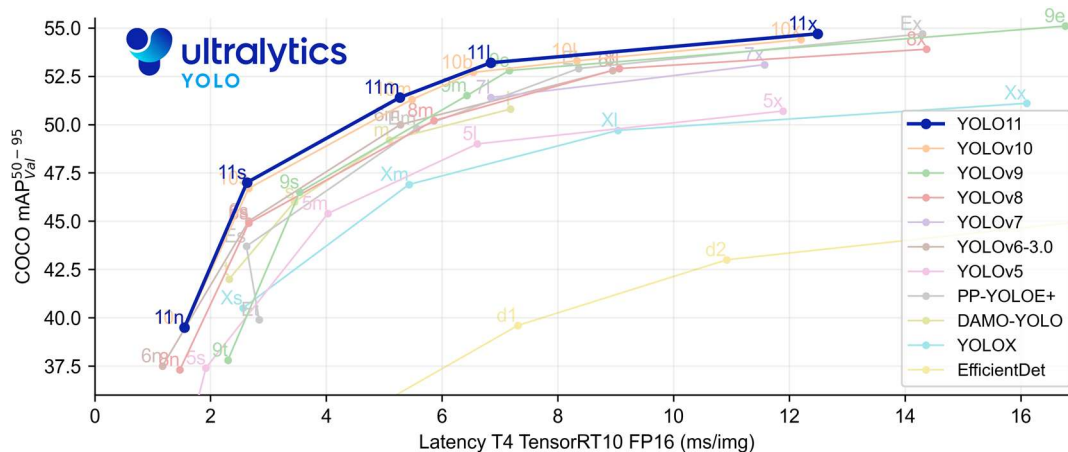


Рисунок 2.5 – Порівняння моделей YOLO за точністю та затримкою [12]

Порівняння моделей YOLO за точністю та затримкою, що наочно продемонстровано на рисунку 2.5, підкреслює її переваги. Її здатність обробляти

складні сцени з малими об'єктами, перекриттям і шумними фонами, а також швидкість і точність роблять YOLOv11 оптимальним вибором для системи.

## 2.5 Вибір бібліотек

Бібліотеки програмування є наборами попередньо розроблених функцій, класів і модулів, які значно спрощують створення програмного забезпечення, надаючи готові інструменти для вирішення складних задач. Вони дозволяють розробникам економити час, уникаючи написання коду з нуля, і забезпечують високу надійність завдяки перевіреним рішенням, створеним спільнотою експертів. У комп'ютерному зорі бібліотеки відіграють ключову роль, надаючи можливості для обробки зображень, тренування нейронних мереж, детекції об'єктів і управління великими наборами даних. Наприклад, вони дозволяють створювати системи розпізнавання облич у смартфонах, аналізувати медичні зображення чи керувати автономними транспортними засобами. Поза комп'ютерним зором бібліотеки застосовуються в таких галузях, як аналіз даних, штучний інтелект, робототехніка та автоматизація, забезпечуючи масштабування проєктів і швидку реалізацію складних алгоритмів. У задачах, подібних до класифікації одягу з використанням YOLOv11, бібліотеки необхідні для підготовки даних, реалізації моделей і обробки контенту в реальному часі.

### 2.5.1 Ultralytics

Бібліотека Ultralytics є провідним інструментом для реалізації моделей YOLO, які широко застосовуються в задачах комп'ютерного зору, зокрема для детекції об'єктів, сегментації та класифікації. Вона використовується в автономних транспортних засобах для виявлення пішоходів і транспортних засобів, у системах безпеки для моніторингу загроз, у ритейлі для аналізу поведінки покупців і в сільському господарстві для оцінки стану рослин. Ultralytics підтримує повний цикл роботи з моделями YOLO, включаючи тренування, передбачення, експорт і

					ІК13.240БАК.006 ПЗ	Арк.
Зм.	Лист	№ докум.	Підпис	Дата		28

оцінку, а також забезпечує оптимізацію для GPU і edge-пристроїв. Ultralytics вирізняється простотою використання, дозволяючи навіть новачкам швидко налаштувати модель, і підтримує оптимізацію для високої продуктивності через інноваційні архітектури у YOLOv11.

### 2.5.2 Roboflow

Roboflow – це платформа для створення, анотації та управління датасетами, яка застосовується в задачах комп’ютерного зору, таких як розпізнавання об’єктів, сегментація та аналіз зображень. Вона використовується в медичній діагностиці для підготовки даних для виявлення аномалій, у промислових системах для контролю якості, а також у спортивній аналітиці для відстеження рухів. Roboflow API автоматизує анотацію (обмежувальні рамки, мітки класів), аугментацію (зміна яскравості, обертання, шум) і експорт даних у форматах, сумісних із YOLO, PyTorch чи іншими фреймворками. Бібліотека спрощує створення якісних датасетів, що підвищує точність моделей у специфічних задачах.

### 2.5.3 PyTorch

PyTorch – це бібліотека для створення й тренування нейронних мереж, яка широко застосовується в академічних дослідженнях, промислових проєктах і розробці штучного інтелекту. Вона використовується для комп’ютерного зору (детекція об’єктів, класифікація зображень), обробки природної мови (мовні моделі) і робототехніки (автономне керування). PyTorch підтримує динамічні обчислювальні графи, що полегшують експерименти, і забезпечує оптимізацію для GPU, прискорюючи обробку великих даних [15]. Бібліотека є основою для багатьох сучасних моделей, таких як YOLO, ResNet чи Transformers, і дозволяє адаптувати архітектури до специфічних задач. Її гнучкість та простота використання роблять PyTorch відмінним вибором для швидкого прототипування та складних

дослідницьких завдань, а широка спільнота розробників постійно вдосконалює функціонал та надає підтримку.

#### 2.5.4 OpenCV

Бібліотека OpenCV є універсальним інструментом для обробки зображень і відео, який застосовується в комп'ютерному зорі, робототехніці, медичній візуалізації та промисловій автоматизації. Вона використовується для виявлення об'єктів у системах безпеки, аналізу медичних зображень (наприклад, МРТ), розпізнавання жестів у геймінгу та обробки даних із дронів. OpenCV підтримує операції зміни розміру, нормалізації, фільтрації, сегментації та конвертації форматів, забезпечуючи швидку обробку великих обсягів даних. Бібліотека оптимізована для роботи з різними платформами і сумісна з Python, що робить її популярною в задачах реального часу.

#### Висновок до розділу 2

У другому розділі було проведено всебічний аналіз і обґрунтування вибору технологій для створення системи з автоматичною класифікацією одягу. На основі аналізу технічних викликів визначено ключові етапи розробки: вибір мови програмування, вебфреймворку, моделі комп'ютерного зору та бібліотек.

Для реалізації системи обрано мову програмування Python завдяки її універсальності, простому синтаксису та потужній екосистемі бібліотек, що підтверджується лідерством у TIOBE Index (25.35%) [13]. Python забезпечує швидке прототипування й інтеграцію з інструментами глибокого навчання, що критично для обробки зображень із соціальних мереж. Вебфреймворк Django обрано для створення надійного та безпечного інтерфейсу, який дозволяє користувачам завантажувати фото чи відео до 100 МБ і отримувати анотовані результати, перевершуючи Flask і FastAPI за інтегрованістю й простотою розробки.

					ІК13.240БАК.006 ПЗ	Арк.
Зм.	Лист	№ докум.	Підпис	Дата		30

Модель YOLOv11 обрана як основа системи завдяки її високій точності, швидкості і здатності розпізнавати багато категорій одягу у складних сценах. YOLOv11 перевершує класичні CNN (VGG16, ResNet), які не підтримують локалізацію кількох об'єктів, а також двоступеневу Faster R-CNN (mAP 73.2%, 7 FPS) і одноступеневу SSD (mAP 76.8%, 19 FPS) за швидкістю й універсальністю. Її архітектура, оптимізована через C3k2, SPPF і C2PSA, забезпечує ефективну обробку на edge-пристроях і хмарних платформах. В табл. 2.1 наведено порівняння цих моделей.

Таблиця 2.1 – Порівняння ефективності моделей

Модель	Переваги	Недоліки
Класичні CNN (VGG, ResNet)	Висока точність для одного об'єкта	Не локалізують кілька об'єктів
Faster R-CNN	Висока точність у складних сценаріях	Низька швидкість, складна реалізація
SSD	Швидкість, простота	Нижча точність для малих об'єктів
YOLOv11	Висока швидкість і точність	Вимагає якісних даних для тренування

Бібліотеки Ultralytics, Roboflow, PyTorch і OpenCV обрано для реалізації повного циклу роботи системи. Ultralytics забезпечує розгортання YOLOv11, Roboflow автоматизує підготовку кастомного датасету DeepFashion, PyTorch підтримує тренування й адаптацію моделі, а OpenCV обробляє зображення та відео, адаптуючи їх до вимог моделі. Ці інструменти, сумісні з Python, широко застосовуються в комп'ютерному зорі, від автономних транспортних засобів до медичної діагностики, що підтверджує їхню надійність.

Таким чином, було ретельно відібрано оптимальний стек технологій, що є фундаментом для створення високопродуктивної та масштабованої системи. Запропонована методологія забезпечує створення системи, яка ефективно розпізнає кілька елементів одягу на одному зображенні, відповідаючи маркетинговим задачам аналізу трендів.

## 3 РОЗРОБЛЕННЯ НЕЙРОННОЇ МЕРЕЖІ

### 3.1 Підготовка даних для тренування моделі

Підготовка якісного набору даних є критичним етапом розробки нейронної мережі, оскільки від різноманітності, точності анотацій і попередньої обробки залежить ефективність моделі YOLOv11 у задачах класифікації одягу. У рамках цього проєкту створено власний датасет, який містить 3220 зображень із людьми в різноманітному одязі, різних позах і умовах освітлення. Зображення відображають реальні сценарії вуличної моди та контенту соціальних мереж, що відповідає маркетинговим задачам аналізу модних трендів. Датасет охоплює 12 категорій одягу, адаптованих до потреб аналізу повсякденного гардеробу.

#### 3.1.1 Вибір і структура датасету

Для тренування нейронної мережі створено власний датасет, що складається з зображень, зібраних із реальних сцен, які включають людей у різноманітному одязі, різних позах і умовах освітлення. Зображення відображають типові сценарії соціальних мереж та вуличної моди, що робить їх придатними для маркетингових задач, таких як аналіз популярності певних категорій одягу чи стилів. Датасет охоплює 12 категорій одягу, обраних з урахуванням їхньої поширеності у повсякденному гардеробі та різноманітності форм: штани (pants), футболки (Tshirt), спідниці (skirt), сукні (dress), худі (hoodie), взуття (shoe), светри (sweater), толстовки (sweatshirt), куртки (jacket), сорочки (shirt), топи (top), шорти (short). Вибір цих категорій обґрунтовано їхньою актуальністю для аналізу модних трендів, а також потенціалом для застосування результатів класифікації в електронній комерції.

Кожне зображення в датасеті містить одного або кількох людей із видимими елементами одягу, що можуть перекриватися або бути частково прихованими через пози чи фон. Це забезпечує високу реалістичність даних, але водночас створює

					ІК13.240БАК.006 ПЗ	Арк.
						32
Зм.	Лист	№ докум.	Підпис	Дата		

виклики для анотації та класифікації, які вирішуються на етапах обробки та тренування. Приклади зображень наведені на рисунку 3.1.

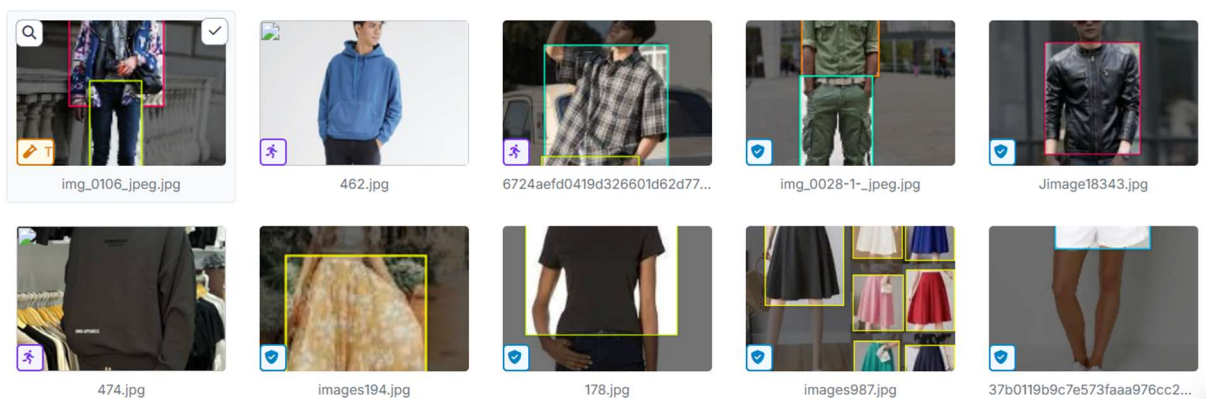


Рисунок 3.1 – Анотовані зображення у інтерфейсі Roboflow

Порівняно з великими датасетами, такими як DeepFashion2 (~800 000 зображень), власний датасет є меншим, але відкритий та завдяки невеликим обсягам даних дає можливість проводити навчання моделі навіть на персональних комп'ютерах або ноутбуках із середніми характеристиками GPU. Крім того, власний датасет легко масштабувати та адаптувати під специфічні завдання, оскільки його розмітка виконувалася вручну, що дозволяє швидко вносити зміни або додавати нові класи одягу.

Розподіл зображень за класами та вибірками (тренувальна, валідаційна, тестова) наведено у таблиці 3.1. Дані розподілено з урахуванням необхідності забезпечення достатньої кількості прикладів для навчання, оцінки гіперпараметрів і тестування моделі. Тренувальна вибірка становить приблизно 65% (2093 зображення), валідаційна – 27% (870 зображень), тестова – 8% (257 зображень). Такий розподіл відповідає стандартним практикам у комп'ютерному зорі, де більша частина даних використовується для тренування, щоб модель могла узагальнювати закономірності, а валідаційна та тестова вибірки дозволяють оцінити її продуктивність.

Таблиця 3.1 – Розподіл зображень датасету за класами та вибірками

Клас	Загальна кількість	Тренувальна	Валідаційна	Тестова
Штани (pants)	571	343	185	43
Футболка (Tshirt)	495	320	125	50
Спідниця (skirt)	431	244	137	50
Сукня (dress)	392	243	110	39
Худі (hoodie)	399	259	95	45
Взуття (Shoe)	384	230	124	30
Светр (sweater)	345	207	104	34
Толстовка (sweatshirt)	317	206	85	26
Куртка (jacket)	307	184	92	31
Сорочка (shirt)	308	200	85	23
Топ (top)	309	185	93	31
Шорти (short)	288	172	86	30
Всього	3220	2093	870	257

Як видно з таблиці 3.1, датасет має певну незбалансованість класів: категорія «штани» містить найбільшу кількість зображень (571), тоді як «шорти» – найменшу (288). Ця незбалансованість може вплинути на точність класифікації менш представлених класів, таких як шорти, топи чи сорочки. Для вирішення цього виклику на етапі створення датасету буде використано аугментацію даних, яка генерує додаткові зображення для кожного класу.

### 3.1.2 Анотація через Roboflow

Анотація зображень становить основу підготовки даних для тренування моделі YOLOv11, оскільки якість і точність позначок визначають здатність моделі розпізнавати та класифікувати об'єкти в реальних умовах. У рамках цього проєкту анотацію всіх зображень датасету виконано вручну за допомогою платформи

					ІК13.240БАК.006 ПЗ	Арк.
Зм.	Лист	№ докум.	Підпис	Дата		34

Roboflow. Цей інструмент обрано завдяки зручному графічному інтерфейсу, підтримці форматів, сумісних із YOLOv11, і можливостям автоматизації перевірки анотацій, що забезпечують високу якість даних.

Процес анотації передбачав створення обмежувальних рамок навколо кожного елемента одягу на зображенні з подальшим присвоєнням відповідної мітки класу. Наприклад, на зображенні людини в шортах, футболці та кедах створювалися три окремі рамки, позначені як «short», «Tshirt» і «Shoe». Усі зображення спочатку завантажувалися до Roboflow із локального сховища у форматі JPEG. Далі за допомогою графічного інтерфейсу платформи створювалися прямокутні рамки, які охоплювали об'єкти з мінімальним включенням фону. Приклад графічного інтерфейсу можна побачити на рисунку 3.2.

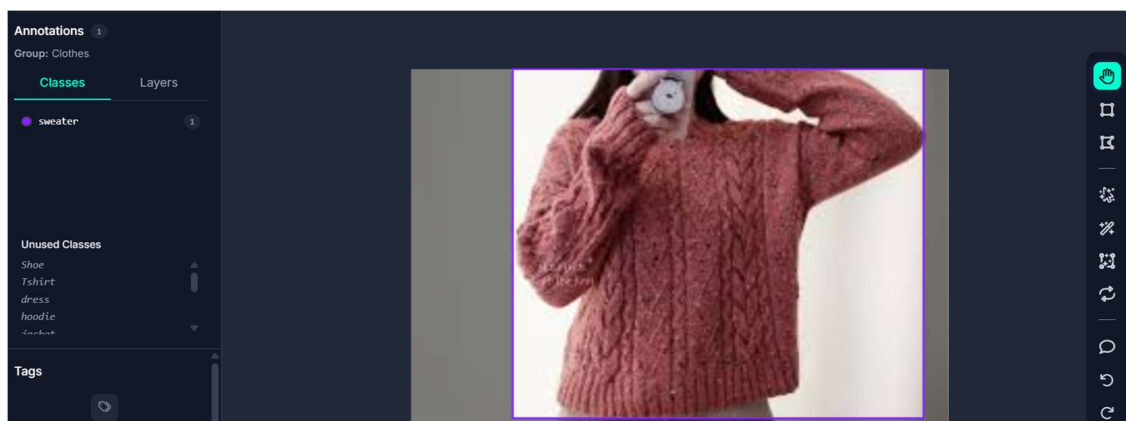


Рисунок 3.2 – Інтерфейс для анотації Roboflow

У результаті отримано повний набір анованих зображень із точними обмежувальними рамками та мітками, готовими для подальшої обробки та експорту.

### 3.1.3 Формування та експорт датасету через Roboflow

Після завершення анотації зображень наступним етапом стало формування структурованого датасету, готового для тренування моделі YOLOv11, та його експорт у відповідному форматі. Цей процес охоплював організацію зображень і

анотацій, застосування попередньої обробки та аугментації, а також підготовку даних для інтеграції в тренувальний процес. Для стандартизації зображень і забезпечення їхньої сумісності з вимогами YOLOv11 у Roboflow застосовано попередню обробку. Зокрема, використано виправлення орієнтації зображень, щоб усунути помилки, пов'язані з неправильним поворотом, наприклад, перевернутими фото. Усі зображення приведено до розміру 640x640 пікселів із збереженням пропорцій, що відповідає вхідному формату моделі. Переведення зображень у відтінки сірого зменшило вплив кольорових варіацій і підвищило стійкість до змін освітлення. Адаптивна еквалізація контрасту покращила чіткість контурів одягу, особливо в зображеннях із низьким контрастом, таких як темний одяг на темному фоні. Ці методи обробки підвищили якість даних, адаптувавши їх до реальних умов.

Щоб компенсувати незбалансованість класів, наприклад, 288 зображень для шортів проти 571 для штанів, і підвищити стійкість моделі до варіацій одягу, поз чи освітлення, у Roboflow застосовано аугментацію даних. Для кожного зображення тренувальної вибірки генерувалося три додаткові версії з такими трансформаціями: зміна експозиції від -10% до +10%, поворот від -15° до +15°, зміна яскравості від -15% до +15% і зміна відтінку від -11° до +11%. Це дозволило збільшити ефективний розмір тренувальної вибірки з 2093 до приблизно 6279 зображень. Наприклад, для класу «шорти» з 172 тренувальними зображеннями створено близько 516 зображень, що значно підвищило різноманітність даних для цього класу. Аугментація враховувала реальні сценарії, такі як різні кути зйомки чи освітлення, сприяючи кращому розпізнаванню рідкісних класів.

Після виконання анотації, попередньої обробки та аугментації датасет експортується з платформи Roboflow у форматі, сумісному з моделлю YOLOv11, із подальшим завантаженням у робочу директорію для тренування моделі.

### 3.2 Конфігурація моделі

					ІК13.240БАК.006 ПЗ	Арк.
						36
Зм.	Лист	№ докум.	Підпис	Дата		





Параметр	YOLOv11n	YOLOv11s
Терпимість (patience)	10	10
Мозаїчна аугментація	1.0	1.0
Горизонтальне віддзеркалення	0.5	0.5
Кількість класів	12	12
Базова модель	YOLOv11n (COCO)	YOLOv11s (COCO)

Конфігурації YOLOv11n та YOLOv11s забезпечують ефективну роботу моделей для класифікації 12 категорій одягу. Використання трансферного навчання та налаштування гіперпараметрів дозволило підготувати моделі до подальшого навчання, яке розглядається у наступному підрозділу.

### 3.3 Тренування нейронної мережі

Для тренування моделей YOLOv11n та YOLOv11s використовувалося апаратне забезпечення із графічним процесором NVIDIA (8 ГБ відеопам'яті) на локальній машині. Це дало змогу ефективно обробляти великі обсяги даних та скоротити час тренування порівняно з використанням центрального процесора.

Робота розпочалася з імпорту датасету в робочу директорію у форматі YOLO, що включав зображення та анотації для класів одягу. Код імпорту датасету в робочу директорію показаний на рисунку 3.3. Дані були розподілені на навчальну, валідаційну та тестову множини.

```
# Завантаження датасету з Roboflow
rf = Roboflow(api_key= )
project = rf.workspace(" ").project(" ")
version = project.version(7)
dataset = version.download("yolov11")
```

Рисунок 3.3 – Код імпорту датасету в робочу директорію

Для прискорення адаптації використано попередньо навчені ваги COCO. YOLOv11n обрано для швидшого навчання, YOLOv11s – для підвищення точності. Ініціалізація проводилася через файли yolov11n.pt і yolov11s.pt.

Процес тренування охоплював до 100 епох для обох моделей, але механізм ранньої зупинки з терпимістю 10 епох завершив навчання YOLOv11n на 50-й епосі, а YOLOv11s – на 46-й, базуючись на попередньому експериментальному аналізі. Такий підхід запобіг перенавчанню, зупиняючи процес за відсутності покращення метрик. Основними показниками якості слугували mAP@0.5, precision і recall, які відстежувалися протягом усього циклу.

Для YOLOv11n mAP@0.5 зростає з 0.301 (1 епоха) до 0.716 (50 епох), для YOLOv11s – з 0.399 (1 епоха) до 0.762 (46 епох). Стабілізація mAP@0.5 після 40–50 епох для YOLOv11n і 40–46 епох для YOLOv11s підтверджує ефективність ранньої зупинки. Характеристики тренування наведено в таблиці 3.5.

Таблиця 3.5 – Характеристики тренування моделей YOLOv11n і YOLOv11s

Параметр	YOLOv11n	YOLOv11s
mAP@0.5 (1 епоха)	0.301	0.399
mAP@0.5 (кінцева)	0.716 (50 епох)	0.762 (46 епох)
Precision (кінцева)	0.694	0.782
Recall (кінцева)	0.667	0.695

Результати навчання оцінюються в підпункті 3.4.

### 3.4 Оцінка ефективності моделі

Після завершення тренування проведено оцінювання ефективності моделей YOLOv11n та YOLOv11s на тестовій вибірці власного датасету. Основними метриками оцінки були:

- mAP@0.5 – середня точність при порозі IoU = 0.5;
- precision – точність класифікації;
- recall – частка виявлених об’єктів серед усіх наявних;



Модель демонструє високу точність класифікації для класів hoodie (0.86) та shorts (0.85), що свідчить про її здатність надійно розпізнавати ці об'єкти у більшості випадків. Проте для класу shoe точність класифікації становить лише 0.36, що вказує на дві ключові проблеми: по-перше, ймовірно, недостатня кількість тренувальних зразків цього класу у датасеті, а по-друге, висока схожість взуття з фоном.

Окрему увагу заслуговують також класи hoodie (0.86) та jacket (0.55), для яких модель досягає порівняно високої точності, але спостерігається часте плутання між цими класами.

Таку ж проблему мають класи dress (0.76) та skirt (0.71), що зумовлено їх схожістю в структурі. Це свідчить про необхідність подальшого вдосконалення якості анотацій та аугментаційних методів для підвищення роздільної здатності між подібними класами.

Варто також відзначити, що клас background має значний відсоток помилкових передбачень, коли модель плутає фонові області з об'єктами, зокрема shoe (0.64). Це може бути наслідком недостатньої кількості негативних прикладів у тренувальній вибірці або викликано слабким контрастом між об'єктами та фоном.

Проаналізуємо нормалізовану матрицю плутанини для YOLOv11s.

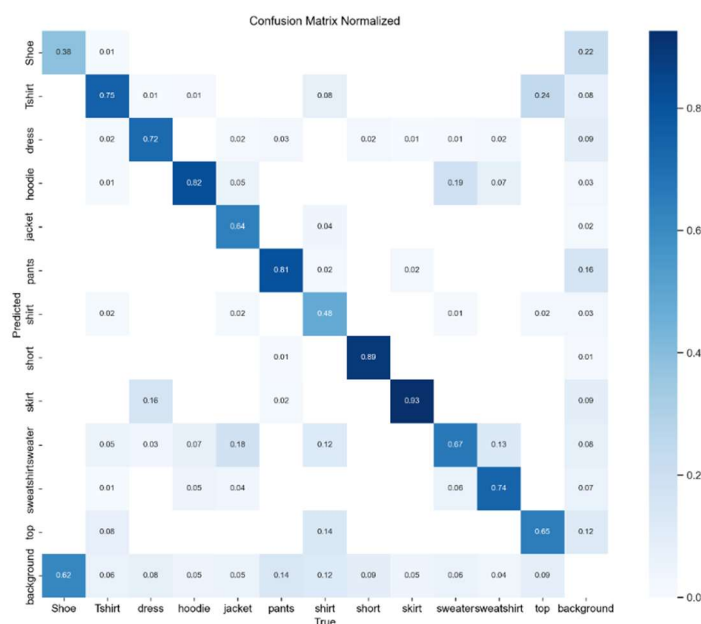


Рисунок 3.5 – Нормалізована матриця плутанини для YOLOv11s

На рисунку 3.5 наведено матрицю плутанини для YOLOv11s, яка відображає розподіл точності за класами: shoe (0.36), t-shirt (0.75), dress (0.72), hoodie (0.82), pants (0.81), shorts (0.89), jacket (0.64), skirt (0.93), sweater (0.67), sweatshirt (0.74), shirt (0.48) і top (0.65). Порівняно з YOLOv11n, модель YOLOv11s демонструє помітне покращення точності для класів jacket, skirt, sweater і sweatshirt, що свідчить про ефективнішу адаптацію до цих категорій. Водночас клас shirt втратив у точності (0.48), хоча зменшення плутанини з іншими класами вказує на стабільнішу класифікацію. Клас top (0.65) характеризується як зниженням точності, так і підвищеною плутаниною з іншими категоріями. Проблема схожості між dress і skirt, а також між hoodie і sweater зберігається, що відображає складність розрізнення об'єктів із подібною структурою. Загалом, модель YOLOv11s демонструє тенденцію до плутанини переважно між класами з аналогічними структурними ознаками, що зменшує помилки в широкому спектрі категорій.

### 3.5 Тестування моделі

Після завершення тренування моделей YOLOv11n і YOLOv11s та їхньої оцінки за метриками, було проведено серію тестів для перевірки їхньої працездатності в реальних умовах. Основною метою було оцінити швидкість обробки, точність класифікації та стійкість до складних сцен, зокрема різних поз людей і схожих класів одягу. Для цього використано 100 випадкових зображень із тестової вибірки та додатковий набір із 10 нових зображень, які не входили до тренувальних даних, щоб перевірити здатність моделей до узагальнення.

#### 3.5.1 Швидкість роботи

Для тестування моделей було обрано випадкові 100 зображень із тестової множини власного датасету. Кожне зображення подавалося на вхід моделей YOLOv11n та YOLOv11s, після чого для кожної моделі фіксувався час обробки та порівнювалися прогнозовані класи із реальними анотаціями. Тестування

проводилося на CPU, щоб штучно підкреслити різницю у швидкості роботи моделей. Для проведення тестування використовувався код, наведений у Додатку 2, який автоматизує процес оцінки: завантажує зображення, виконує інференс для кожної моделі, зберігає передбачені класи та обчислює час обробки.

Одним із важливих критеріїв ефективності моделі є час, необхідний для обробки одного зображення. У таблиці 3.6 наведено порівняння часу інференсу для обох моделей.

Таблиця 3.6 – Порівняння часу інференсу YOLOv11n та YOLOv11s

Модель	Середній час інференсу на зображення (с)	Сумарний час інференсу на 100 зображень (с)
YOLOv11n	0.1711	17.11
YOLOv11s	0.2707	27.07

За результатами тестування модель YOLOv11n продемонструвала середній час інференсу 0.1711 секунди на одне зображення та сумарний час 17.11 секунд на всю вибірку. Натомість YOLOv11s мала середній час 0.2707 секунди та сумарний час 27.07 секунд. Отримані результати свідчать про те, що YOLOv11n працює помітно швидше, що пояснюється меншою кількістю параметрів та меншою обчислювальною складністю моделі.

### 3.5.2 Аналіз складних випадків

Під час тестування особливу увагу було приділено здатності моделей коректно класифікувати об'єкти одягу у випадках, коли на зображеннях люди перебували у різних позах, а також при наявності схожих за візуальними ознаками категорій одягу.

На рисунку 3.6 показано приклад із дівчиною в спідниці, знятої боком. YOLOv11s класифікувала об'єкт як “dress” із упевненістю 0.84, хоча правильною міткою була “skirt”, тоді як YOLOv11n не розпізнала об'єкт (упевненість < 0.5). Це

вказує на кращу здатність YOLOv11s виявляти об'єкти в складних позах, хоч і з помилками в класифікації схожих класів.



Рисунок 3.6 – Порівняння результатів nano та small моделей на спідниці

На рисунку 3.7 показано два приклади різних поз: людина лежить в штанах і на іншому фото стоїть спиною в светрі та штанах. Обидві моделі продемонстрували гарні результати в розпізнаванні об'єктів. YOLOv11s досягла упевненості 0.85 для штанів лежачої людини та 0.80 для светра і 0.82 для штанів стоячої людини, тоді як YOLOv11n показала упевненість 0.60 для штанів лежачої людини та 0.65 для светра і штанів стоячої. Це свідчить про високу адаптивність обох моделей до складних поз, із дещо вищою упевненістю YOLOv11s.

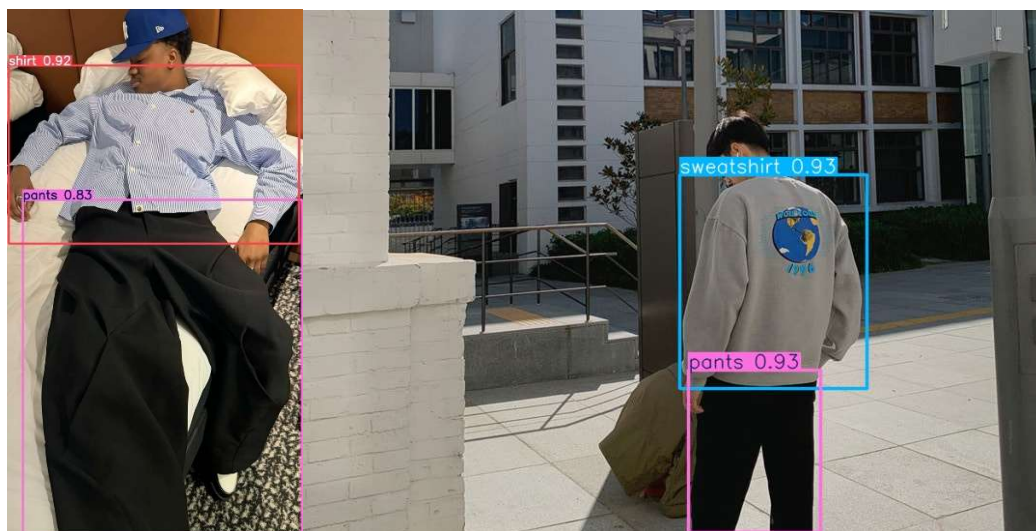


Рисунок 3.7 – Приклади успішних класифікацій складних випадків

Зм.	Лист	№ докум.	Підпис	Дата

### Висновки до розділу 3

У третьому розділі було розглянуто повний цикл розробки, тренування та оцінювання нейронних мереж YOLOv11n та YOLOv11s для задачі класифікації одягу на основі власного датасету. Здійснено підготовку даних, у тому числі анотацію, аугментацію та попередню обробку, що дозволило забезпечити достатню різноманітність та реалістичність тренувальної вибірки. Проведено конфігурацію моделей YOLOv11n та YOLOv11s, що відрізняються кількістю параметрів та обчислювальною складністю, а також адаптацію їх до специфіки завдання.

У процесі тренування використано механізм ранньої зупинки (early stopping), який дозволив уникнути перенавчання: тренування YOLOv11n завершено на 50-й епосі, а YOLOv11s – на 46-й епосі. Це рішення базувалося на аналізі метрик mAP@0.5, precision і recall, що стабілізувалися у відповідні моменти навчання. Згідно з результатами, YOLOv11n продемонструвала середню точність (mAP@0.5) 0.716, precision 0.694 та recall 0.667, тоді як YOLOv11s досягла mAP@0.5 0.762, precision 0.782 та recall 0.695.

Тестування моделей на 100 зображеннях показало, що YOLOv11n має середній час інференсу 0.1711 с, що робить її оптимальною для завдань із високими вимогами до швидкості, тоді як YOLOv11s демонструє 0.2707 с із вищою точністю. Аналіз складних сцен, таких як лежача людина в штанах і стояча людина в светрі та штанах, підтвердив високу адаптивність обох моделей. Порівняльний аналіз моделей засвідчив, що YOLOv11n краще підходить для швидких обчислень за умови обмежених ресурсів, тоді як YOLOv11s демонструє вищу точність на більшості класів, особливо тих, що мають складну структуру або високий рівень варіативності. Матриці плутанини для обох моделей вказують на проблеми розрізнення близьких за ознаками класів (наприклад, dress/skirt, hoodie/sweater), а також на складнощі при класифікації класу shoe через його візуальну схожість із фоном.

Незважаючи на досягнуті високі показники, подальше підвищення точності та надійності системи потребує вирішення виявлених проблем із

					ІК13.240БАК.006 ПЗ	Арк.
						46
Зм.	Лист	№ докум.	Підпис	Дата		

незбалансованістю класів та схожістю візуальних ознак. Застосовані методи аугментації даних частково компенсували ці недоліки, проте додаткові стратегії, такі як *oversampling* для менш представлених категорій або вдосконалення архітектури моделі для розрізнення тонких візуальних відмінностей, можуть бути корисними.

Отримані результати підкреслюють важливість якості анотацій та збалансованості класів у тренувальній вибірці. Розроблені моделі показують потенціал для впровадження в реальні застосунки, зокрема в системи автоматизованої класифікації одягу. У подальших дослідженнях доцільно зосередитися на поліпшенні якості даних, зокрема за рахунок збільшення кількості зразків для рідкісних класів і вдосконалення аугментаційних технік.

З огляду на кращі результати та вищу загальну точність, для подальшого використання у проєкті обрано модель YOLOv11s. Її буде конвертовано у формат ONNX для подальшого використання в продуктивному середовищі. Це дозволило зберегти оптимальну архітектуру моделі та забезпечити сумісність із широким спектром інструментів, зокрема ONNX Runtime Web, що дає змогу виконувати інференс безпосередньо у браузері користувача без потреби у серверній обробці. Завдяки цьому модель буде інтегрована у клієнтську частину системи, де здійснюватиме класифікацію одягу в реальному часі.

					ІК13.240БАК.006 ПЗ	Арк.
						47
Зм.	Лист	№ докум.	Підпис	Дата		

## 4 РОЗРОБЛЕННЯ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

### 4.1 Архітектура системи

Розроблена система реалізована як вебзастосунок WebClothesClassification, який працює за принципом «клієнт–сервер». Такий підхід дозволяє розподілити навантаження між частинами, що відповідають за обробку даних (сервер), і частинами, що забезпечують взаємодію з користувачем (клієнт). Це забезпечує ефективне використання ресурсів та дозволяє легко інтегрувати різні функціональні модулі. Також робить систему гнучкою, масштабованою та зручною для подальшого розвитку.

Система підтримує роботу з фотографіями, відеофайлами та онлайн-відеотрансляціями (через вебкамеру), дозволяючи користувачу завантажувати фото чи відео для аналізу або спостерігати результати класифікації у режимі реального часу. Для роботи використано Django (Python), який забезпечує зручну організацію коду та шаблонів для створення динамічного інтерфейсу користувача. Завдяки використанню HTML5, CSS3 і JavaScript, користувачі можуть комфортно працювати із завантаженням медіа та переглядати результати у вигляді зображень із підсвіченими елементами одягу або таблиць із статистикою.

Загальний сценарій взаємодії користувача з системою представлено на ІК13.240БАК.006 Д1, де відображені ключові етапи: завантаження медіафайлів, обробка на сервері та відображення результатів в інтерфейсі користувача.

### 4.2 Основні компоненти системи

Система складається з кількох важливих компонентів, організованих у зрозумілу та логічну структуру каталогів, як показано на рисунку 4.1. Ця структура ретельно продумана для забезпечення модульності, легкості розширення та спрощення навігації по проекту, що є ключовим для ефективної розробки та підтримки.

					ІК13.240БАК.006 ПЗ	Арк.
						48
Зм.	Лист	№ докум.	Підпис	Дата		

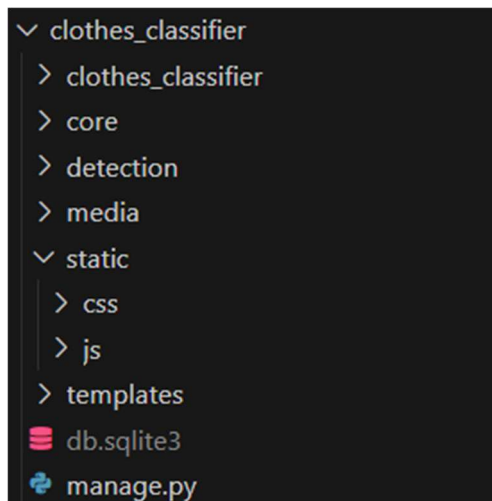


Рисунок 4.1 – Структура проєкту

У кореневій директорії знаходиться папка `clothes_classifier`, де зберігаються головні налаштування системи, зокрема файли конфігурації та маршрутизації запитів. У папці `core` розміщена основна логіка обробки: тут реалізовані скрипти для завантаження фотографій, відео та потокової трансляції через вебкамеру. Також саме тут розташовані шаблони інтерфейсу, серед яких: `home.html` – головна сторінка для завантаження медіа; `result.html` – сторінка для відображення результатів класифікації фото; `video_result.html` – сторінка для перегляду результатів класифікації відео; `stream.html` – сторінка для онлайн-трансляції з вебкамери. Приклад головної сторінки продемонстрований на рисунку 4.2.

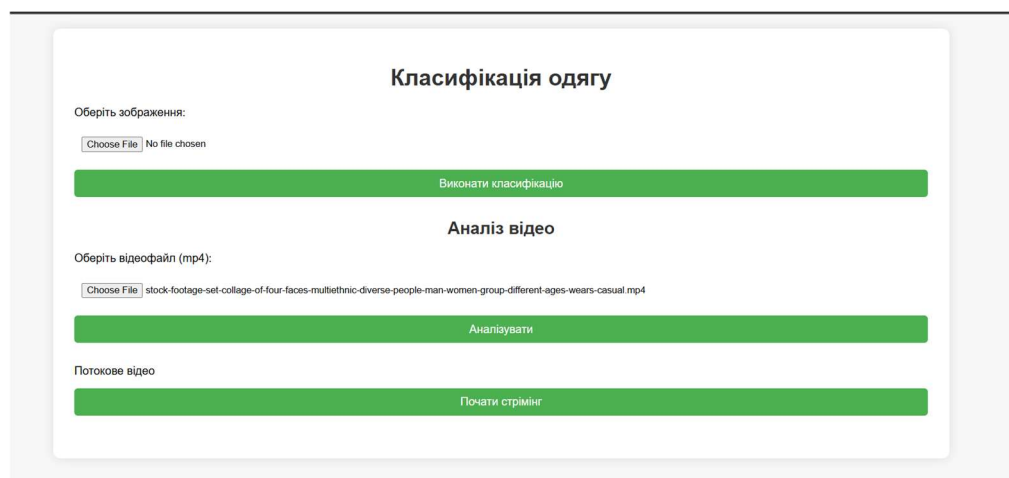


Рисунок 4.2 – Головна сторінка

Усі ці шаблони наслідують від base.html, що є особливістю архітектури Django та дозволяє використовувати спільні стилі, заголовки та навігаційні панелі, забезпечуючи єдину візуальну стилістику та спрощуючи підтримку інтерфейсу.

Крім цього, у папці static знаходяться файли стилів (CSS) та скрипти (JavaScript), які відповідають за візуальне оформлення й інтерактивність користувацького інтерфейсу. Окрема папка detection містить необхідні ресурси для реалізації комп'ютерного зору, серед яких: best.pt – навчена модель YOLO для розпізнавання одягу, inference.py – скрипт для обробки фото, video.py – скрипт для обробки відео.

### 4.3 Взаємодія між компонентами

Система працює за принципом взаємодії клієнта та сервера, де користувач надсилає фото, відео або підключає вебкамеру через інтерфейс вебзастосунку, а серверна частина відповідає за їхню обробку та повернення результатів. Такий підхід дозволяє обробляти зображення та відео у реальному часі або близько до реального, що робить використання системи зручним та інтерактивним.

Послідовність обробки даних, починаючи від завантаження користувачем медіа і закінчуючи поверненням результатів класифікації, наочно представлена на діаграмі потоку даних ІК13.240БАК.006 ДЗ. Це дозволяє побачити, як система організовує обмін даними між модулями й забезпечує швидкий зворотний зв'язок.

Після завантаження фото користувачем через вебсторінку файл передається на сервер, де він зберігається у спеціальній директорії. Далі фото передається до модуля комп'ютерного зору, де за допомогою моделі YOLO виконується класифікація. На цьому етапі визначаються координати кожного виявленого об'єкта, його клас (тип одягу) та рівень впевненості в результаті. Після обробки система повертає користувачу результат у вигляді фото з нанесеними рамками та підписами, а також текстовим або графічним відображенням результатів на сторінці. Приклад виводу результату для фотографії можна побачити на рисунку 4.3.

					ІК13.240БАК.006 ПЗ	Арк.
						50
Зм.	Лист	№ докум.	Підпис	Дата		

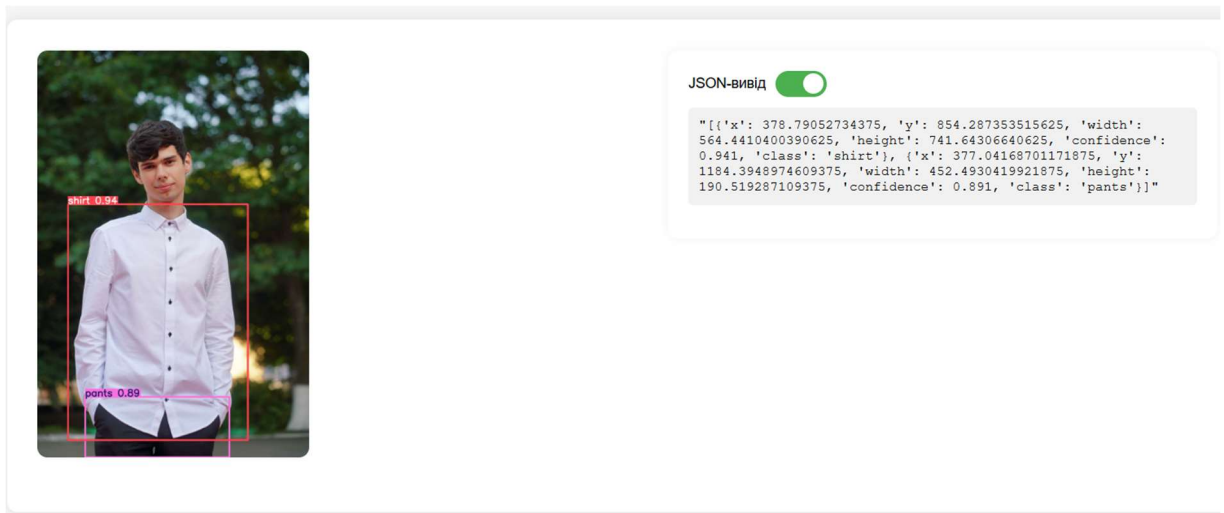


Рисунок 4.3 – Вивід результату для фото

У випадку з відеофайлом користувач завантажує його через вебінтерфейс. Система приймає відео та зберігає його на сервері, після чого запускається покадровий аналіз із використанням моделі класифікації. Кожен кадр аналізується окремо, для нього визначаються класи одягу та підраховується кількість появ на різних часових ділянках. На основі цього формується інтерактивний звіт, що відображає статистику появи кожного типу одягу протягом усього відео, а також генерується новий відеофайл з нанесеними анотаціями. Ці результати користувач бачить у вебінтерфейсі разом із таблицею присутності класів одягу, як продемонстровано на рисунку 4.4.

**Результати аналізу відео**

JSON-вивід

Відсоток присутності кожної речі:

Клас	Загальна присутність (%)	По таймфреймах (кожні 10 сек)
short	16.30	0–10с: 16.30%
Shoe	0.74	0–10с: 0.74%
Tshirt	20.00	0–10с: 20.00%

Рисунок 4.4 – Вивід результату для відео

Для трансляції у режимі реального часу використовується локальна обробка на пристрої користувача. Для забезпечення класифікації одягу користувачам надається модель YOLO, конвертована в формат ONNX.

Це дозволяє користувачу підключити свою вебкамеру через браузер, а попередньо завантажена легка версія моделі проводить класифікацію одягу безпосередньо на його пристрої, не потребуючи надсилання відеопотоку на сервер. Це суттєво зменшує затримку та залежність від мережевого з'єднання, а також значно знижує навантаження на серверні ресурси. Після першого відкриття сторінки модель зберігається локально на пристрої у IndexedDB, що усуває необхідність повторного завантаження при наступних відвідуваннях, підвищуючи швидкість запуску та автономність роботи. IndexedDB ідеально підходить для цього завдяки своїй здатності ефективно обробляти великі бінарні файли, такі як модель, забезпечуючи надійне зберігання з лімітом до 1 ГБ та підтримку асинхронного доступу, що дозволяє уникати блокування основного потоку браузера під час роботи з даними. Саме IndexedDB обрано замість кешування чи інших методів зберігання, оскільки воно пропонує більший обсяг пам'яті (порівняно з лімітом 5-10 МБ у localStorage), кращу організацію даних через підтримку баз даних і ключ-значення, а також можливість асинхронного доступу, що є критичним для швидкої обробки великих файлів без переривання роботи браузера. Завантаження моделі показано на рисунку 4.5.

```
// Завантаження моделі ONNX із IndexedDB
let session;
async function loadModel() {
  const dbName = 'modelDB';
  const storeName = 'models';
  const request = indexedDB.open(dbName, 1);

  request.onupgradeneeded = (event) => {
    const db = event.target.result;
    db.createObjectStore(storeName);
  };

  request.onsuccess = async (event) => {
    const db = event.target.result;
    const transaction = db.transaction(storeName, 'readwrite');
    const store = transaction.objectStore(storeName);
    const cachedModel = await new Promise(resolve) => {
      const request = store.get('best.onnx');
      request.onsuccess = () => resolve(request.result);
    };
  };
}
```

Рисунок 4.5 – Завантаження моделі

Крім цього, передається також коротка статистика щодо присутності одягу, яку можна динамічно відобразити на екрані для покращення користувацького досвіду. Приклад виводу результату для трансляції представлено на рисунку 4.6.

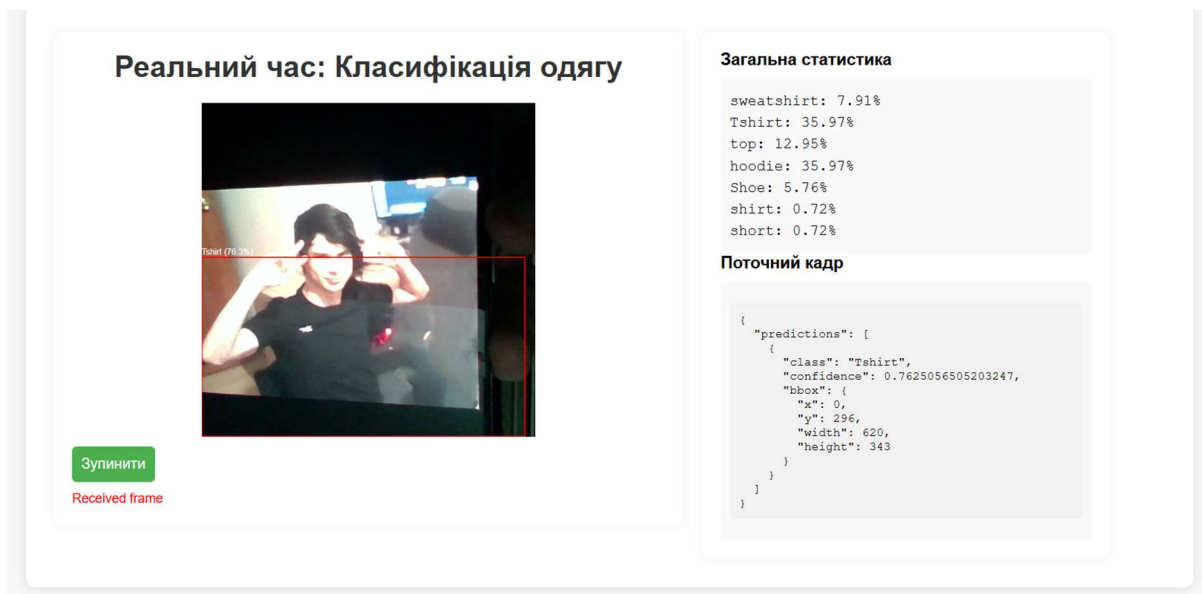


Рисунок 4.6 – Вивід результату для трансляції

#### 4.4 Особливості архітектури та переваги

Архітектура системи побудована таким чином, щоб забезпечити зручну та ефективну обробку мультимедійних даних, зокрема фото, відео та потокового відео з вебкамери. Для цього використовується архітектурний патерн MVT (Model-View-Template), притаманний фреймворку Django. Завдяки цьому патерну забезпечується чіткий поділ логіки додатку на окремі складові: Model (структура даних та взаємодія з базою, якщо потрібно), View (логіка обробки запитів) та Template (відображення результатів для користувача). Це спрощує розробку, полегшує внесення змін та пришвидшує налагодження системи.

Особливістю архітектури є використання бібліотеки ONNX.js у поєднанні з Django, що дозволяє виконувати класифікацію одягу локально на пристрої користувача за допомогою моделі YOLO, конвертованої у формат ONNX. Модель передається клієнту один раз через HTTP-запит і завантажується як статичний

ресурс із сервера. Це усуває потребу в постійному з'єднанні або WebSocket, зменшує навантаження на сервер та забезпечує приватність користувацьких зображень.

Після завантаження модель використовується для обробки кадрів із вебкамери або завантажених зображень безпосередньо в браузері. Інференс виконується за допомогою ONNX.js на стороні клієнта, що підвищує автономність системи та дозволяє користувачу отримувати анотації одягу у режимі реального часу. Такий підхід також сприяє масштабовності системи, дозволяючи підтримувати одночасну роботу з великою кількістю користувачів без значного збільшення серверних ресурсів. Крім того, використання ONNX.js забезпечує гнучкість, дозволяючи легко оновлювати модель або адаптувати її до нових класів одягу без зміни серверної інфраструктури.

Взаємодію між модулями системи та розподіл даних між ними наведено на діаграмі послідовності ІК13.240БАК.006 Д2. Ця діаграма дозволяє наочно зрозуміти, як організовано обмін даними між користувацьким інтерфейсом, серверною логікою та модулем комп'ютерного зору.

#### Висновки до розділу 4

У четвертому розділі було розглянуто особливості розробки програмного забезпечення системи WebClothesClassification, що дозволяє автоматизувати процес класифікації одягу на фото, відео та у режимі реального часу. Було описано архітектуру системи, яка реалізована на основі клієнт-серверного підходу з використанням фреймворку Django та патерну MVT, що забезпечує чіткий розподіл відповідальностей між компонентами та спрощує розробку і підтримку.

Основні компоненти системи структуровано у вигляді окремих папок для налаштувань, логіки обробки запитів, шаблонів інтерфейсу та ресурсів комп'ютерного зору, що дозволяє легко масштабувати функціонал та інтегрувати нові можливості.

					ІК13.240БАК.006 ПЗ	Арк.
						54
Зм.	Лист	№ докум.	Підпис	Дата		

Було детально описано реалізацію обробки в реальному часі, яка базується на завантаженні моделі YOLO, конвертованої у формат ONNX, і виконанні інференсу за допомогою бібліотеки ONNX.js безпосередньо в браузері. Модель передається користувачу один раз через HTTP-запит як статичний ресурс і зберігається локально після першого відкриття, що усуває потребу в постійному з'єднанні та забезпечує автономну обробку кадрів із вебкамери або завантажених зображень.

Особливу увагу приділено тому, що такий підхід дозволяє системі забезпечувати миттєву класифікацію одягу на стороні клієнта, підвищуючи швидкість відгуку та зменшуючи навантаження на сервер. Ця технологія є особливо актуальною для робототехнічних систем, де автономна обробка зображень у реальному часі, наприклад, для розпізнавання одягу роботами-асистентами чи автономними агентами, може значно підвищити їхню ефективність і незалежність від зовнішніх серверів.

Таким чином, розроблена система відповідає сучасним вимогам до інтерактивності, масштабованості та зручності використання, а її архітектура забезпечує можливість подальшого розширення та інтеграції з іншими сервісами.

					ІК13.240БАК.006 ПЗ	Арк.
Зм.	Лист	№ докум.	Підпис	Дата		55

## 5 РОБОТОТЕХНІЧНЕ ЗАБЕЗПЕЧЕННЯ

### 5.1 Інтеграція з робототехнічними системами

Система класифікації одягу має значний потенціал для інтеграції з робототехнічними платформами, що дозволяє автоматизувати процеси у різних сферах та підвищувати ефективність роботи. Така інтеграція відкриває нові можливості використання системи не лише як програмного забезпечення для класифікації одягу, а й як компонента у складних роботизованих системах.

Однією з основних сфер застосування є електронна комерція, де система може автоматично визначати та тегувати товари за категоріями одягу у великих каталогах інтернет-магазинів. Це значно спрощує навігацію для користувачів та підвищує якість організації товарів. Крім того, систему класифікації одягу можна інтегрувати з роботизованими складами для автоматизованого сортування одягу за його типами, як на рисунку 5.1, що оптимізує роботу логістичних центрів. Це можливо завдяки тому, що навчена модель не лише визначає клас об'єкта, а й повертає координати його розташування у кадрі, що дозволяє точно локалізувати одяг та передавати ці дані виконавчим модулям роботизованої платформи.

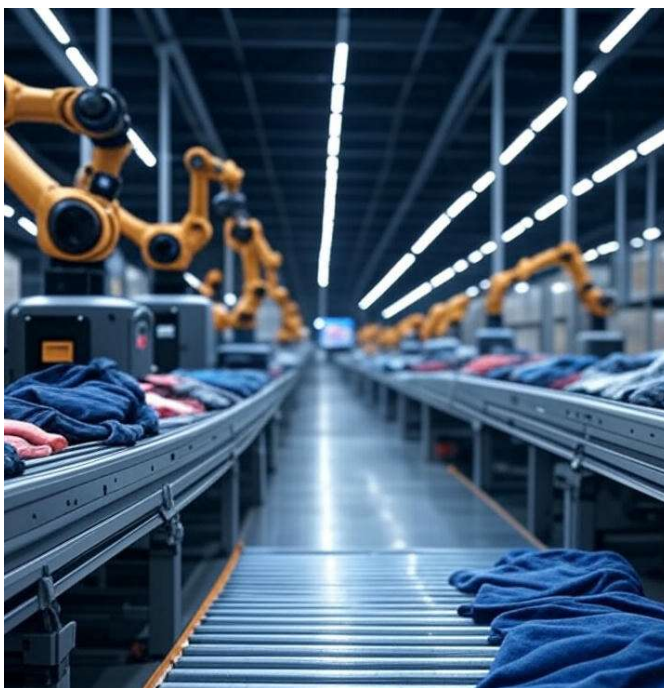


Рисунок 5.1 – Приклад роботизованого складу

Зм.	Лист	№ докум.	Підпис	Дата

У модній індустрії система класифікації одягу демонструє значний потенціал для інтеграції з роботизованими системами збору даних із соціальних мереж, для аналізу модних трендів. Роботизовані платформи, оснащені камерами та обчислювальними модулями, можуть автономно збирати статистику присутності речей, зображення та відео зі стрімів чи заходів для подальшої обробки або покращення моделі. Цей процес використовується для моніторингу популярності різних категорій одягу та формування деталізованих рекомендацій для виробників і ритейлерів, які базуються на статистичних даних, таких як частота появи певних категорій одягу. Такий підхід дозволяє прогнозувати асортимент до поточних уподобань споживачів, оптимізувати виробничі цикли та підвищувати конкурентоспроможність на ринку.

У сфері безпеки система класифікації одягу може бути інтегрована з відеоспостереженням для ідентифікації людей за одягом. Це особливо актуально для пошуку людей за описом їхнього одягу у великих скупченнях або на масових заходах, зокрема в ситуаціях, коли обличчя людини не видно, що робить ідентифікацію за одягом особливо цінною. Такий підхід звужує коло пошуку, дозволяючи оперативно визначити потенційні збіги на основі типів одягу, таких як куртки, шапки чи штани. Додатково система може бути інтегрована у концепцію «розумного міста» для моніторингу громадських просторів, де роботизовані платформи виконують патрулювання, а модуль класифікації збирає та аналізує дані про типи одягу.



Рисунок 5.2 – Приклад роботи для патрулювання

## 5.2 Апаратна частина робототехнічного забезпечення

Для повноцінної інтеграції системи класифікації одягу із робототехнічними платформами важливою складовою є апаратне забезпечення. Воно забезпечує отримання вхідних даних, передачу відеопотоків та безпосередню взаємодію з користувачем або іншими системами.

Основною складовою є камери та сенсори, які можуть бути реалізовані у вигляді вебкамер, IP-камер або вбудованих модулів робототехнічних платформ. Вони забезпечують передачу відеопотоку у реальному часі, що є ключовим для коректної роботи системи. Вибір типу камери залежить від конкретного сценарію використання. Для онлайн-моніторингу чи застосування у примірочних кімнатах можуть використовуватись стандартні вебкамери, тоді як для промислових застосувань доцільніше використовувати IP-камери з вищою роздільною здатністю та кращою стабільністю роботи.

Також важливу роль відіграє обчислювальний модуль, який виконує локальну обробку відеопотоку з камери. Вибір модуля залежить від типу завдання: для нединамічних завдань, таких як сортування одягу, достатньо менш потужного рішення, як Raspberry Pi 4, яке може впоратися з інференсом моделі YOLO, конвертованої в формат ONNX. Для динамічних завдань, таких як відстеження заходів чи патрулювання у реальному часі, рекомендуються потужніші модулі: NVIDIA Jetson Nano або Jetson Xavier NX завдяки окремому GPU, що прискорює інференс. Ці платформи підтримують headless браузеры, такі як Chromium у режимі без графічного інтерфейсу та сумісні з альтернативними методами отримання даних із вебсторінки, наприклад, через Puppeteer чи Selenium, або бібліотеки типу Playwright для парсингу вебданих. Такий підхід дозволяє обробляти дані безпосередньо на пристрої, зменшуючи залежність від зовнішніх серверів і підвищуючи автономність системи.

Ще одним важливим елементом є інтерфейси підключення, які забезпечують передачу даних до серверу. Це можуть бути дротові інтерфейси для стабільної роботи в стаціонарних умовах або бездротові технології для мобільних

					ІК13.240БАК.006 ПЗ	Арк.
Зм.	Лист	№ докум.	Підпис	Дата		58

робототехнічних систем. Крім того, для мобільних застосувань дедалі частіше використовуються мобільні мережі, такі як 5G, які забезпечують високу пропускну здатність і низьку затримку, що ідеально підходить для передачі відеопотоку в реальному часі. Вибір інтерфейсу залежить від вимог до пропускну здатності, стабільності з'єднання та умов експлуатації системи, а також від наявності покриття мобільної мережі в конкретному середовищі.

### 5.3 Передача та обробка даних

У системі особливу увагу приділено інтеграції механізмів передачі та обробки даних у контексті робототехнічних платформ, де необхідно забезпечити стабільну та безперервну роботу в режимі реального часу. Роботи, оснащені камерами або сенсорами, збирають фото та відео, передаючи їх на сервер через безпечне з'єднання, наприклад, VPN/Proxy. Ці дані розміщуються в черзі для подальшої обробки на ML-сервері, де модель YOLO, завантажена на сервері з GPU, виконує класифікацію або може донавчатись, адаптуючи систему до нових умов або класів одягу. Такий підхід дозволяє централізовано оновлювати модель на основі зібраних даних. Водночас автономна класифікація об'єктів у реальному часі здійснюється безпосередньо на клієнті за допомогою моделі YOLO, що працює в браузері. Ця локальна обробка дозволяє роботам аналізувати відеопотік із камер без постійного з'єднання з сервером, збираючи статистику про виявлені типи одягу. Розгортку такої системи зображено на кресленику ІК13.240БАК.006 Д4. Така автономність є критичною для сценаріїв сортування одягу, де кожен кадр забезпечує актуальну інформацію для прийняття рішень, а також для моніторингу.

### Висновки до розділу 5

У п'ятому розділі було розглянуто особливості робототехнічного забезпечення системи класифікації одягу та його роль у забезпеченні ефективної роботи системи у складі робототехнічних платформ. Було детально описано, як

					ІК13.240БАК.006 ПЗ	Арк.
Зм.	Лист	№ докум.	Підпис	Дата		59

система може інтегруватися із робототехнічними рішеннями для автоматизації процесів у різних сферах: електронній комерції, модній індустрії та безпеці.

Розглянуто апаратну частину, яка включає камери, сенсори, сервери обробки даних та інтерфейси зв'язку, що дозволяють інтегрувати систему у різноманітні сценарії використання. Особливу увагу приділено механізмам передачі та обробки даних, де роботи збирають фото та відео для донавчання моделі на локальному сервері, а також виконують автономну класифікацію об'єктів у реальному часі за допомогою ONNX.js, забезпечуючи стабільну роботу та збір статистики.

Важливим аспектом є також синхронізація компонентів та управління у режимі реального часу, що дозволяє системі ефективно працювати навіть у складних умовах інтеграції з мобільними чи стаціонарними роботизованими платформами. Окремо було підкреслено переваги використання робототехнічного забезпечення: підвищення рівня автоматизації, зниження навантаження на користувача та можливість роботи у важкодоступних умовах, що робить систему універсальною та придатною для впровадження у різних галузях.

Таким чином, система класифікації одягу має усі необхідні інструменти та архітектурні рішення для інтеграції з робототехнічними платформами, що значно розширює її функціональні можливості та підвищує ефективність роботи у режимі реального часу.

## ВИСНОВКИ

У дипломному проєкті було розроблено інтелектуальну роботизовану систему класифікації одягу, яка дозволяє автоматично визначати типи одягу на фото та відео з людьми у режимі реального часу. Основною метою роботи було створення універсальної та масштабованої системи, що здатна ефективно розпізнавати одяг у різних сценаріях застосування, зокрема в електронній комерції, модній індустрії та системах безпеки.

Виконаний аналіз існуючих рішень показав, що більшість із них мають обмежену функціональність, зокрема у випадках перекриття об'єктів чи складних фонових умов. Для подолання цих недоліків було обрано сучасну модель YOLOv11, яка забезпечує високу швидкість та точність обробки кадрів завдяки своїй архітектурі та підтримці багатозадачності. Завдяки використанню бібліотек Ultralytics, Roboflow, PyTorch та OpenCV було досягнуто ефективного функціонування системи як у стаціонарних, так і в мобільних умовах.

Система реалізована у вигляді вебзастосунку на базі фреймворку Django з використанням архітектурного патерну MVT, що дозволяє розділити логіку додатку на окремі складові та полегшує його підтримку та масштабування. Для реалізації режиму реального часу використовуються сучасні вебтехнології, зокрема формат моделі ONNX у поєднанні з ONNX.js, що дає змогу виконувати інференс безпосередньо в браузері на боці клієнта, що є особливо важливим для інтеграції з робототехнічними платформами.

Перспективи розвитку системи полягають у її подальшому вдосконаленні та розширенні функціональності. Зокрема, можливим є впровадження модулів відстеження брендів одягу для покращення каталогізації товарів у інтернет-магазинах та аналітики ринку, що також сприятиме покращенню ідентифікації людей. Додатково, перспективним напрямом є реалізація модуля визначення поз користувачів, що дозволить більш точно локалізувати та ідентифікувати елементи одягу у складних сценах завдяки використанню просторових та геометричних

					ІК13.240БАК.006 ПЗ	Арк.
Зм.	Лист	№ докум.	Підпис	Дата		61

пріорів. Також можливе розроблення модуля ідентифікації кольорів, текстур та матеріалів для глибшого аналізу модних трендів та рекомендацій користувачам.

У розділі 5 було розглянуто можливість інтеграції системи з робототехнічними платформами. Система може бути використана у логістичних центрах для автоматичного сортування одягу, у модній індустрії для аналізу трендів, а також у системах відеоспостереження для ідентифікації людей за одягом. Також було розглянуто питання апаратного забезпечення, особливості передачі та обробки даних у реальному часі та синхронізації роботи між компонентами системи.

Основні результати проекту повністю відповідають завданню дипломного проекту та вимогам до інтелектуальних роботизованих систем. Розроблена система демонструє високу точність, надійність та інтерактивність. Отримані результати дозволяють впроваджувати систему у різних сферах, зокрема у сфері електронної комерції, модної індустрії та безпеки, що відкриває широкі перспективи для її практичного застосування.

Таким чином, проект підтвердив актуальність тематики, досягнув поставленої мети та створив підґрунтя для подальших досліджень і вдосконалення системи класифікації одягу у різноманітних реальних умовах.

## ПЕРЕЛІК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. He, K., Zhang, X., Ren, S., Sun, J. Deep residual learning for image recognition // Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. – 2016. – С. 770–778.
2. Hara, K., Jagadeesh, V., Piramuthu, R. Fashion apparel detection: The role of deep convolutional neural networks // CVPR Workshops. – 2016. – С. 1–8.
3. Bedeli M., Geradts Z., van Eijk E. Clothing identification via deep learning: Forensic applications // Forensic Sciences Research. – 2018. – Vol. 3, № 3. – P. 219–229.
4. Ge, Y., Zhang, R. DeepFashion2: A benchmark for clothing analysis // CVPR. – 2019. – С. 1–10.
5. Szegedy, C., Vanhoucke, V., Ioffe, S., Shlens, J., Wojna, Z. Rethinking the Inception architecture for computer vision // CVPR. – 2016. – С. 2818–2826.
6. Ge, Y., Zhang, R. DeepFashion2: A benchmark for clothing analysis // CVPR. – 2019. – С. 1–10.
7. Liu, Z., Luo, P., Qiu, S., Wang, X., Tang, X. DeepFashion: Powering robust clothes recognition // CVPR. – 2016. – С. 1096–1104.
8. Yamaguchi, K., Berg, T. L., Ortiz, L. E. Chic or social: Visual popularity analysis in online fashion networks // ACM Multimedia. – 2014. – С. 773–776.
9. Instagram. URL: <https://about.instagram.com>.
10. TikTok and Artificial Intelligence: The Intersection of Social Media and Cutting-Edge Technology. URL: <https://community.ibm.com/community/user/ibmcommunity/blogs/paul-glenn2/2024/09/13/tiktok-and-artificial-intelligence-the-intersectio>
11. Pinterest. Ai at Pinterest. URL: <https://help.pinterest.com/en/article/ai-at-pinterest>
12. Ultralytics YOLO11. URL: <https://docs.ultralytics.com/models/yolo11/>
13. TIOBE. TIOBE Index. URL: <https://www.tiobe.com/tiobe-index/>

14. OpenCV Team. OpenCV: Open Source Computer Vision Library. URL: <https://opencv.org>
15. PyTorch Team. PyTorch: An Open Source Machine Learning Framework. URL: <https://pytorch.org>
16. VGG-16 | CNN model. URL: <https://www.geeksforgeeks.org/vgg-16-cnn-model/>
17. Ren S., He K., Girshick R., Sun J. Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks // arXiv preprint arXiv:1506.01497. – 2015. – DOI: 10.48550/arXiv.1506.01497.
18. Sanchez S. A., Romero H. J., Morales A. D. A review: Comparison of performance metrics of pretrained models for object detection using the TensorFlow framework // IOP Conference Series Vol. 844. – 012024. – DOI: 10.1088/1757-899X/844/1/012024.
19. FastAPI. FastAPI: A Modern, Fast (High-Performance), Web Framework. URL: <https://fastapi.tiangolo.com>
20. Python Async (ASGI) Web Frameworks Benchmark. URL: <https://klen.github.io/py-frameworks-bench/>