

## РЕФЕРАТ

Дипломна робота: 83 сторінок, 39 рисунків, 12 таблиць, 12 джерел.

Ціль роботи – огляд методів класифікації об'єктів на зображення на основі згорткових нейронних мереж.

В роботі розглянуті існуючі методи класифікації об'єктів на зображенні. Представлені нейронні мережі детектування об'єктів: Fast R-CNN, Faster R-CNN, YOLOv3, YOLOv5, RetinaNet. Розглянуті основні метрики оцінки якості та реалізовані скрипти. Проведено детектування об'єктів на тестовому датасеті COCO.

МЕТРИКИ, ЗГОРТКОВІ НЕЙРОННІ МЕРЕЖІ, YOLOV3, YOLOV5, FAST R-CNN, FASTER R-CNN, ЯКІР, ШАР.

## ABSTRACT

Thesis: 83 pages, 39 figures, 12 tables, 12 references.

The aim of the work is to review the methods of classification of image objects on the basis of initial neuron networks.

In the study of existing methods of classification of image objects. Neural networks of object detection are presented: Fast R-CNN, Faster R-CNN, YOLOv3, YOLOv5, RetinaNet. The basic metrics of quality assessment and implemented scripts are considered. Detection of objects on the test data of SOSO is carried out.

METRICS, CONVOLUTIONAL NEURAL NETWORKS, YOLOV3, YOLOV5, FAST R-CNN, FASTER R-CNN, ANCHOR, LAYER.

## Зміст

ВСТУП .....	7
1. КЛАСИФІКАЦІЯ .....	8
1. Огляд методів класифікації .....	8
1.1.1. Підхід, заснований на даних .....	9
2. Машинне навчання .....	10
3. Огляд мереж .....	11
2. ТЕХНОЛОГІЇ .....	12
1. Аналіз структури нейронних мереж .....	12
2.2.1. Архітектура Fast R-CNN .....	12
2.2.2. Критерії точності .....	12
3. РОЗРОБКА ТА АНАЛІЗ .....	13
3.1. Аналіз точності класифікації .....	13
4. ЕКОНОМІЧНА ЧАСТИНА .....	14
4.1. Визначення основної ідеї та цілей роботи .....	14
4.2. Розрахунок витрат .....	15
ВИСНОВОКИ .....	23
ПЕРЕЛІК ПОСИЛАНЬ .....	24

## ВСТУП

В сучасному світі задачі детектування об'єктів та тема комп'ютерного зору є дуже актуальними. З розвитком технологій та науки методи класифікації ставали все кращими та показували відмінні результати. Після появи глибокого навчання та згорткових нейронних мереж показники точності та швидкості детектування стали на порядок кращими. Сьогодні існує велика кількість підходів до машинного навчання, виявлення та класифікації об'єктів.

За допомоги машинного навчання розробнику не обов'язково писати інструкції, які би включали в себе всі можливі проблеми та містили всі рішення. Замість цього в комп'ютер чи програму закладають алгоритм самостійного знаходження рішень шляхом комплексного використання статистичних даних, з яких виводяться закономірності і на основі яких робляться прогнози.

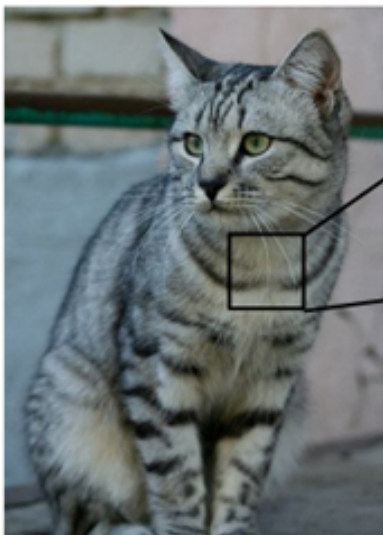
В цілі даної дипломної роботи входить аналіз вже існуючих нейронних мереж, створення інтерфейсів мереж для оцінки якості їх роботи та можливості використовувати в подальшому. Оцінка роботи нейронної мережі дає змогу зрозуміти, чи є доцільним використовувати ту чи іншу мережу в певній сфері. Можна вибрати, що для вас головніше велика швидкість детектування чи точність, або знайти баланс між двома показниками.

# 1. КЛАСИФІКАЦІЯ

## 1. Огляд методів класифікації

Існує багато підходів, за допомогою яких вирішують задачу пошуку об'єкта на зображенні. Основна проблема при розгляданні задачі класифікації полягає в тому, що сприйняття зображення людиною та машиною суттєво відрізняється. Коли комп'ютер дивиться на зображення, то не бачить цілісне уявлення kota або будь-якого іншого об'єкта, він бачить лише гігантську сітку чисел. Наприклад, якщо розмір зображення 800 на 600 і кожен піксель представлений трьома числами для червоного, зеленого і синього каналів, то вийде матриця яка буде нараховувати 1,440,000 чисел. Дуже важко виділити з них якийсь конкретний предмет (рис. 1.1).

Що бачить людина



[1485 112 188 111 184 99 186 99 96 183 112 119 184 97 93 87]
[ 91 98 182 186 184 79 98 183 99 185 123 136 118 185 94 85]
[ 76 85 98 185 128 185 87 96 95 99 115 112 186 183 99 85]
[ 99 81 81 93 128 131 127 188 95 98 182 99 96 93 181 94]
[186 91 61 64 69 91 88 85 181 187 189 98 75 84 96 95]
[114 188 85 55 55 69 64 54 64 87 112 129 98 74 84 91]
[133 137 147 183 65 81 88 65 52 54 74 84 182 93 85 82]
[128 137 144 148 189 95 86 78 62 65 63 63 68 73 86 181]
[125 133 148 137 119 121 117 94 65 79 88 65 54 64 72 98]
[127 125 131 147 133 127 126 131 111 96 89 75 61 64 72 84]
[115 114 189 123 158 148 131 118 113 189 188 92 74 65 72 78]
[ 89 93 98 97 188 147 131 118 113 114 113 189 186 95 77 88]
[ 63 77 86 81 77 79 182 123 117 115 117 125 125 138 115 87]
[ 62 65 82 89 78 71 88 181 124 126 119 181 187 114 131 139]
[ 63 65 75 88 89 71 62 81 128 138 135 185 81 98 118 118]
[ 87 65 71 87 186 95 69 45 76 138 126 187 92 94 185 112]
[118 97 82 86 117 123 116 66 41 51 95 93 89 95 182 187]
[164 146 112 88 82 128 124 184 76 48 45 66 88 181 182 189]
[157 178 157 128 93 86 114 132 112 97 69 55 78 82 99 94]
[138 128 134 161 139 188 189 118 121 134 114 87 65 53 69 86]
[128 112 96 117 158 144 128 115 184 187 182 93 87 81 72 79]
[123 187 96 86 83 112 153 149 122 189 184 75 88 187 112 99]
[122 121 182 88 82 86 94 117 145 148 153 182 58 78 92 187]
[122 164 148 183 71 56 78 83 93 183 119 139 182 61 69 84]

Що бачить машина

Рисунок 1.1 – Представлення даних

Ця проблема отримала назву «семантичний розрив» – нерозуміння інформації, яка укладена в даних. Наприклад, якщо ми будемо розглядати

зображення на яких кіт буде представлений з іншого ракурсу або при іншому освітленні, то вся сітка чисел буде виглядати зовсім інакше. Крім цього, тварини можуть приймати безліч різних поз, або ж на зображенні може виявитися тільки частина об'єкта. Алгоритми розпізнавання повинні бути стійкі до таких змін об'єктів.

Можна спробувати створити набір правил, які будуть описувати еталон. Наприклад, ми можемо виявити границі об'єкта на зображенні а потім класифікувати різні кути і границі які би описували об'єкт. Але прямі набори правил не дають стабільний результат, будь-яке відхилення від еталона може вносити похибки і для кожного нового об'єкта треба буде створювати новий набір правил. Тому такий підхід не може бути масштабованим.

#### 1.1.1. Підхід, заснований на даних

Замість того, щоб вручну намагатися створити набір правил, можна зібрати великий набір даних з зображеннями певного класу (кішок, літаків, оленів та інших речей). Для цього можна використати пошук картинок Google або вже готовий датасет. Потім необхідно навчити класифікатор, відправивши в нього всі зібрані зображення. На виході ми отримаємо модель, узагальнюючу знання про розпізнавання різних об'єктів. Після цього вона зможе працювати на нових зображеннях і відрізняти деякі об'єкти.

Отже, замість однієї функції, яка просто розпізнає об'єкт на вхідному зображенні, ми отримали дві: перша функція «навчання», робота якої полягає в обробці зображень і створення моделі. Друга функція «прогноз» розпізнає нові зображення. Разом вони складають основу для згорткових нейронних мереж і глибокого навчання в цілому

## 2. Машинне навчання

У машинному навчанні система навчається, а не програмується явно. Їй передаються численні приклади, які стосуються задачі, а вона знаходить в цих прикладах статистичну структуру, яка дозволяє системі виробити правила для автоматичного виконання завдання (рис. 1.2). Наприклад, щоб автоматизувати завдання визначення фотографій, зроблених у відпустці, можна передати системі машинного навчання безліч прикладів фотографій, вже класифікованих людьми, і система вивчить статистичні правила класифікації конкретних фотографій [1].



Рисунок 1.2 – Машинне навчання: нова парадигма програмування

Для машинного навчання основними трьома складовими є: контрольні вхідні дані, приклади очікуваних результатів, спосіб оцінки якості роботи алгоритму.

Контрольні вхідні дані – наприклад, якщо вирішується завдання розпізнавання мови, такими контрольними вхідними даними можуть бути файли із записом мови різних людей; якщо вирішується завдання класифікації зображень, такими даними можуть бути зображення.

Приклади очікуваних результатів – в задачі розпізнавання мови це можуть бути транскрипції звукових файлів, складені людьми; в задачі класифікації зображень очікуваним результатом можуть бути теги, такі як «собака», «кішка».

### 3. Огляд мереж

Основні нейронні мережі для детектування об'єктів на зображенні які ми розглянемо: Fast R-CNN, Faster R-CNN, YOLOv3, YOLOv5, RetinaNet.



## 2.

## ТЕХНОЛОГІЇ

## 1. Аналіз структури нейронних мереж

Роздивимось більш детально нейронні мережі, їх структуру та основні елементи, які дозволяють ефективно детектувати об'єкти на зображеннях.

## 2.2.1. Архітектура Fast R-CNN

RoI pooling шар приймає на вхід карту особливостей, отриману від останнього згорткового шару нейронної мережі, і RoI претендента (в координатах зображення). RoI перетворюється з координат зображення в координати на карті особливостей і на отриманий прямокутник накладається сітка  $W \times H$  з наперед заданими розмірами. Робиться max pooling по кожному осередку цієї сітки. Таким чином RoI pooling шар перетворює вектор особливостей довільного прямокутника з вихідного зображення в вектор особливостей фіксованої розмірності [3].

## 2.2.2. Критерії точності

Щоб зрозуміти наскільки якісно працює нейронна мережа треба ввести міри оцінювання. Для досліджуваних нейронних мереж візьмемо наступні параметри: матриця помилок (confusion matrix), IoU (перекриття отриманим блоком площі істинного блоку).

## 3.

## РОЗРОБКА ТА АНАЛІЗ

## 3.1. Аналіз точності класифікації

Для оцінки точності будемо використовувати метрику IoU. Скрипт, який буде розраховувати метрику (рис. 3.1):

```
import numpy as np

def __calc_iou(true, predicted):
    if type(true[0]) is list:
        true = true[0]
    xA = max(true[0], predicted[0])
    yA = max(true[1], predicted[1])
    xB = min(true[2], predicted[2])
    yB = min(true[3], predicted[3])
    interArea = max(0, xB - xA + 1) * max(0, yB - yA + 1)
    boxAArea = (true[2] - true[0] + 1) * (true[3] - true[1] + 1)
    boxBArea = (predicted[2] - predicted[0] + 1) * (predicted[3] - predicted[1] + 1)
    iou = interArea / float(boxAArea + boxBArea - interArea)
    return iou

def IoU(y_true, y_predict, consider_empty=False):
    """
    Calculate min, max and mean IoU for predicted data
    bbox_array = {
        '1': [[xmin, ymin, xmax, ymax], [xmin, ymin, xmax, ymax] ...]...
    }
    :param y_true: bbox_array
    :param y_predict: bbox_array
    :param consider_empty: is consider empty bboxes for image
    :return: array of IoU [min, mean, max]
    """
    IoU_array = []
    for image in y_true.keys():
        if image in y_predict:
            for bbox_true in y_true[image]:
                iou_local = []
                for bbox_predict in y_predict[image]:
                    iou_local.append(__calc_iou(bbox_true, bbox_predict))
                iou_local_idx = np.argmax(iou_local)
                IoU_array.append(iou_local[iou_local_idx])
            elif consider_empty:
                IoU_array.extend([0 for _ in y_true[image]])
    return [np.min(IoU_array), np.mean(IoU_array), np.max(IoU_array)]
```

Рисунок 3.1 – Функція метрики IoU

Розглянемо показники метрик для нейронних мереж. Результати визначення IoU занесемо в таблицю 3.1.

Таблиця 3.1 – Показник IoU

Назва мережі	YOLOv3	YOLOv5s	RetinaNet	Faster R-CNN
Розмір батча	Показник IoU			
100	0,54	0,57	0,71	0,69
500	0,48	0,56	0,69	0,67
1000	0,47	0,58	0,70	0,68
5000	0,48	0,55	0,69	0,66
10000	0,49	0,54	0,69	0,66

За результатами видно, що найкращі показники IoU показує нейронна мережа Faster R-CNN.

## 4. ЕКОНОМІЧНА ЧАСТИНА

### 4.1. Визначення основної ідеї та цілей роботи

Ідея цього проекту полягає в дослідженні передових моделей нейронних мереж та аналіз методів класифікації цих мереж. В результаті роботи ми отримаємо якісні характеристики нейронних мереж і виявимо мережі з найкращими показниками точності та швидкодії.

#### 4.2. Розрахунок витрат

Роботу буде виконувати один працівник – молодший інженер Python. Проведемо розрахунок середньої часової ставки виконавця роботи: заробітна плата молодшого розробника складає 14000 гривень на місяць. Враховуємо, що молодший розробник відробляє по 8 годин у день 5 днів у тиждень. В середньому, впродовж місяця 21 робочий день, тобто 168 годин в місяць. Отже, можна розрахувати погодинний оклад працівника. Формула для розрахунку:

$$L = \frac{P}{H}, \quad (4.1)$$

де  $L$  – погодинний оклад робітника, грн;

$P$  – місячна заробітна плата робітника, грн;

$H$  – кількість відпрацьованих годин.

Виходячи з формули (4.1), розрахуємо погодинний оклад молодшого розробника:

$$L = \frac{14000}{168} = 83 \text{ (грн)}.$$

Проаналізуємо перелік необхідних робіт за проектом та визначимо загальну кількість годин на виконання робіт (таблиця 4.1).

Таблиця 4.1 – Перелік робіт

Номер етапу	Назва стадій та етапів	Трудомісткість, люд./год. (Т)
		Молодший розробник
I	Опис нейронних мереж та методів класифікації	50
II	Програмування	50
III	Аналіз роботи нейронних мереж	20

Разом:	120
--------	-----

Відповідно до таблиці. 4.1, трудомісткість дослідження складає 90 люд./год. Основна заробітна плата виконавця роботи розраховується:

$$ЗП_{осн} = L \cdot T, \quad (4.2)$$

де  $T$  – трудомісткість, люд.-год.

$L$  – погодинна зарплата, грн;

Додаткова заробітна плата, яка у даному випадку встановлена на рівні 20% від основної заробітної плати, розраховується за формулою:

$$ЗП_{дод} = ЗП_{осн} \cdot K_{зп.дод}, \quad (4.3)$$

де  $ЗП_{дод}$  – додаткова заробітна плата робітника, грн;

$ЗП_{осн}$  – основна заробітна плата робітника, грн;

$K_{зп.дод}$  – коефіцієнт додаткової зарплати.

Загальна заробітна плата виконавця роботи розраховується за формулою:

$$ЗП_{заг} = ЗП_{осн} + ЗП_{дод}. \quad (4.4)$$

Розрахуємо основну заробітну плату молодшого розробника використовуючи формулу (4.2):

$$ЗП_{осн} = 83 \cdot 120 = 9960 \text{ (грн)}$$

Розрахуємо додаткову заробітну плату молодшого розробника використовуючи формулу (4.3):

$$ЗП_{дод} = 9960 \cdot 0,2 = 1992 \text{ (грн)}$$

Розрахуємо загальну заробітну плату молодшого розробника використовуючи формулу (4.4):

$$ЗП_{\text{заг}} = 9960 + 1992 = 11952 \text{ (грн)}$$

ЄСВ (єдиний соціальний внесок) встановлено на рівні 22% від фонду зарплати, розраховується за формулою:

$$ЄСВ = ЗП_{\text{заг}} \cdot K_{\text{ЄСВ}}, \quad (4.5)$$

де  $K_{\text{ЄСВ}}$  – коефіцієнт єдиного соціального внеску.

Розрахуємо ЄСВ використовуючи формулу (4.5):

$$ЄСВ = 11952 \cdot 0,22 = 2629 \text{ (грн)}$$

Для підготовки необхідних документів і звітів використовувалося різне канцелярське приладдя. Перелік і вартість канцелярських інструментів і витратних матеріалів наведено в табл. 4.2.

Таблиця 4.2 – Вартість канцтоварів

№ п/п	Найменування	Призначення	Кількість, шт.	Ціна за одиницю, грн	Сума, грн
1	Папір офісний Zoom А4 75 г/м2 клас С 500 аркушів білий	Друк документації, звітів, поміток	1	84	84
2	Упаковка матових файлів формату А4 (100 шт/уп)	Зберігання документації, звітів, поміток	1	84	84

3	К а н ц т о в а р и у асортименті	Запис поміток	1	75	75
Разом:			243		

Загальна вартість канцелярських засобів для оформлення та зберігання документації для проекту складає 243 гривні.

В рамках проведення роботи необхідно приміщення для розгортання інфраструктури робочого місця. З цією метою було орендовано приміщення на підприємстві площею 9 кв. м. Приміщення обладнане всіма необхідними комунікаціями, а також доступом до мережі «інтернет» за окрему плату.

Додаткові витрати включають в себе послуги «Інтернет» . Вартість послуги «Інтернет» складає 160 грн/міс, відповідно на день – 5,3 грн. Тривалість проведення роботи складає 120 годин, тобто повних 15 робочих днів. Розрахуємо розмір витрат на оплату послуги «Інтернет»:

$$Vi = 15 * 5,3 = 79,5 \text{ (грн)}$$

Орендна плата за приміщення розраховується за формулою:

$$P_{OP} = P_D \cdot \frac{T}{T_D}, \quad (4.7)$$

де  $P_D$  – орендна плата на день, грн.;

$T$  – трудомісткість, год.;

$T_D$  – годин роботи в трудовому дні, год.

Таким чином, розрахуємо вартість орендної плати за формулою (4.7):

$$350 \cdot \frac{120}{8} = 350 \cdot 15 = 5250 \text{ (грн)}$$

Розрахуємо загальну суму амортизаційних відрахувань (за проект). Вона розраховується по формулі:

$$A_T = \frac{C_{\text{п}} \cdot N_{\text{ам}}}{100\%}, \quad (4.8)$$

де  $C_{\text{п}}$  – первісна вартість обладнання, грн.;

$N_{\text{ам}}$  – річна норма амортизації за видами обладнання, %.

Загальну суму амортизаційних відрахувань відобразимо у табл. 4.3 використовуючи формулу (4.8).

Таблиця 4.3 – Розрахунок річної суми амортизаційних відрахувань.

№ п/п	Обладнання	Кількість	Вартість грн.	Сума грн.	Норма амортизації %	Річні амортизаційні відрахування, грн.
1	Комп'ютер	1	30000	30000	20%	6000
2	Маршрутизатор TP-Link TL-WR841N	1	529	529	20%	105,8
3	Принтер Canon Pixma G2411	1	5000	5000	20%	1000

Розрахуємо величини амортизаційних відрахувань з розрахунку на одну годину роботи обладнання певного виду. Формула для розрахунку:

$$A_H = \frac{A_T}{T}, \quad (4.9)$$



де  $A_T$  – річні амортизаційні відрахування обладнання;

$T$  – річний фонд роботи певного обладнання. ( 1994 години за 2021 рік).

Величини амортизаційних відрахувань з розрахунку на одну годину роботи певного виду обладнання наведені у табл. 4.4, використовуючи формулу (4.9).

Таблиця 4.4 – Розрахунок величини амортизаційних відрахувань з розрахунку на одну годину роботи обладнання.

№ п/п	Обладнання	Річні амортизаційні відрахування, грн.	Річний фонд роботи обладнання, год.	Амортизаційні відрахування за одну годину роботи, грн.
1	Комп'ютер	6000	1994	3
2	Маршрутизатор TP-Link TL-WR841N	105,8	1994	0,05
3	Принтер Canon Pixma G2411	1000	1994	0,50

Опираючись на розрахунки з табл. 4.4, розрахуємо суму амортизації по кожному виду обладнання опираючись на кількість відпрацьованих годин. Формула для розрахунку суми амортизації обладнання виходячи з кількості відпрацьованих годин:

$$A_F = A_H * T, \quad (4.10)$$

де  $A_H$  – амортизаційні відрахування на одну годину роботи обладнання;

$T$  – кількість відпрацьованого обладнанням часу, у год.

Розрахуємо суму амортизації по кожному виду обладнання виходячи з кількості відпрацьованих годин за формулою (4.10). Результати приведені у табл. 4.5.

Таблиця 4.5 – Розрахунок величини амортизаційних відрахувань з розрахунку кількості відпрацьованих годин

№ п/п	Елементи основних фондів	Амортизаційні відрахування за одну годину роботи, грн.	Кількість відпрацьованого часу, год.	Амортизаційні відрахування за відпрацьований час, грн.
1	Комп'ютер	3	120	360
2	Маршрутизатор TP-Link TL-WR841N	0,05	120	6
3	Принтер Canon Pixma G2411	0,50	1	0,50
Разом:				366,5

В результаті обліку всіх перерахованих вище статей витрат можна визначити собівартість дослідження (таблиця 4.6).

Таблиця 4.6 – Собівартість дослідження

№ п/п	Стаття	Сума, грн
1	Заробітна плата молодшого розробника	9960
2	ЄСВ	1992
3	Канцелярське приладдя	243
4	Орендна плата	5250

Продовження таблиці 4.6

№ п/п	Стаття	Сума, грн
5	Плата за Інтернет	79,5
7	Амортизація	366,5
Разом:		17891

Отже, за результатом проведених розрахунків, витрати на аналіз та дослідження нейронних мереж становлять 17891 гривень.

## ВИСНОВОКИ

В дипломній роботі були розглянуті методи класифікації об'єктів на зображенні. Особливу увагу звернули на класифікацію за допомогою нейронних мереж.

У першому розділі розглянули деякі методи класифікації, їх розвиток і вдосконалення. Роздивились основні етапи розвитку машинного навчання, принципи будови нейронних мереж та процес глибокого навчання.

В другому розділі розглянули п'ять нейронних мереж: YOLOv3, YOLOv5, Fast R-CNN, Faster R-CNN, RetinaNet. Розглянули архітектуру мереж, їх документацію та принципи навчання. При аналізі документації R-CNN мереж зробили, що мережа Faster R-CNN використовує майже ті самі технології, що і мережа Fast R-CNN, крім того має вдосконалені частини, тому подальший аналіз мережі Fast R-CNN був недоцільним.

В третьому розділі провели порівняльний аналіз вже чотирьох мереж за допомоги попередньо розроблених програм метрик та інтерфейсів нейронних мереж. З отриманих результатів можна зробити висновок, що мережі YOLO мають велику швидкість детектування (особливо мережа YOLOv5) і при цьому досить непогані показники IoU та точності класифікації. Мережі Faster R-CNN та RetinaNet мають меншу швидкість детектування (більш ніж в два рази), але при цьому точність класифікація вища на 20-25 % ніж у мереж YOLO.

Остання частина, економічний розділ, містить розрахунки заробітної плати, амортизаційні нарахування основних матеріальних фондів та інших додаткових матеріалів в результаті яких отримано собівартість проведення дослідження.

## ПЕРЕЛІК ПОСИЛАНЬ

1. Франсуа Шолле. Глубокое обучение на Python — СПб.: Питер, 2018. — 400 С.: ил. — (Серия «Библиотека программиста»). — ISBN 978-5-4461-0770-4
2. В. Г. Прохоров. Проблемы програмування. — 2008. № 2-3. — Спеціальний випуск 2008 — с. 669-674 — ISSN 1727-4907.
3. Fast R-CNN / [Електронний ресурс]. — Режим доступу: URL: <https://arxiv.org/abs/1504.08083> (дата звернення 30.03.21)
4. Faster R-CNN / [Електронний ресурс]. — Режим доступу: URL: <https://arxiv.org/abs/1506.01497> (дата звернення 30.03.21)
5. YOLOv3 / [Електронний ресурс]. — Режим доступу: URL: <https://arxiv.org/abs/1804.02767> (дата звернення 30.03.21)
6. YOLOv5 / [Електронний ресурс]. — Режим доступу: URL: <https://github.com/ultralytics/yolov5> (дата звернення 30.03.21)
7. RetinaNet / [Електронний ресурс]. — Режим доступу: URL: <https://arxiv.org/abs/1708.02002v2> (дата звернення 30.03.21)
8. MS COCO / [Електронний ресурс]. — Режим доступу: URL: <https://cocodataset.org/#home> (дата звернення 30.03.21)
9. PyTorch / [Електронний ресурс]. — Режим доступу: URL: <https://arxiv.org/abs/1912.01703> (дата звернення 30.03.21)
10. PyTorch и TensorFlow: отличия и сходства фреймворков / [Електронний ресурс]. — Режим доступу: URL: <https://neurohive.io/ru/tutorial/pytorch-vs-tensorflow/>
11. Матрица ошибок (Confusion Matrix) / [Електронний ресурс]. — Режим доступу: URL: <https://www.helenkapatsa.ru/matritsa-oshibok/>