

**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

«\_\_» \_\_\_\_\_ 20\_\_ р.

**Дипломний проєкт  
на здобуття ступеня бакалавра  
за освітньо-професійною програмою «Інформаційне забезпечення  
робототехнічних систем»  
спеціальності 126 «Інформаційні системи та технології»  
на тему: «Застосунок для формування наборів даних та навчання  
нейронних мереж»**

Виконав (-ла):

студент (-ка) IV курсу, групи ІК-93

Пустовий Денис Дмитрович \_\_\_\_\_

Керівник:

Старший викладач кафедри ІСТ

Коваль Олександр Сергійович \_\_\_\_\_

Рецензент:

с.н.с., к. т. н., доцент

Писаренко Юлія Валеріївна \_\_\_\_\_

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

Студент (-ка) \_\_\_\_\_

Київ – 2023 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

«\_\_» \_\_\_\_\_ 20\_\_ р.

**ЗАВДАННЯ**

**на дипломний проєкт студенту**

**Пустовому Денису Дмитровичу**

1. Тема проєкту «Застосунок для формування наборів даних та навчання нейронних мереж», керівник проєкту Коваль Олександр Сергійович, старший викладач кафедри ІСТ, затверджені наказом по університету від «31» травня 2023 р. № 2101-с
2. Термін подання студентом проєкту: 12 червня 2023 року
3. Вихідні дані до проєкту: Практичні навички роботи з нейронними мережами протягом 4 років навчання, завершені проєкти, пов'язані з нейронними мережами, базові навички програмування на Python і вміння працювати з Qt.
4. Зміст пояснювальної записки: аналіз предметної області та постановка задачі, методичне та інструментальне забезпечення, опис програмної реалізації, робота користувача з додатком, висновки.
5. Перелік графічного матеріалу (із зазначенням обов'язкових креслеників, плакатів, презентацій тощо): Діаграма прецедентів (А3), Діаграма бізнеспроцесів BPMN (А3), Діаграма DFD (А3), Діаграма переходу інтерфейсів ч.1 (А3), Діаграма переходу інтерфейсів ч.2 (А3), Діаграма переходу інтерфейсів ч.3 (А3)

6. Дата видачі завдання: 1 лютого 2023 року

### Календарний план

№ з/п	Назва етапів виконання дипломного проєкту	Термін виконання етапів проєкту	Примітка
1	Вибір теми дипломного проєкту	20.01.2023	
2	Аналіз предметної області	10.02.2023	
3	Аналіз наявних рішень та проблем	24.02.2023	
4	Огляд методів розробки десктоп-додатків	03.03.2023	
5	Розробка алгоритму роботи рішення	17.03.2023	
6	Проектування архітектури	31.03.2023	
7	Реалізація спроектованого рішення	28.04.2023	
8	Тестування створеного рішення	05.05.2023	
9	Оформлення дипломного проєкту	12.06.2023	

Студент

Денис ПУСТОВИЙ

Керівник

Олександр КОВАЛЬ

## АНОТАЦІЯ

Пустовий Д.Д. Застосунок для формування наборів даних та навчання нейронних мереж. КПІ ім. Ігоря Сікорського, Київ, 2023.

Проект містить 66 с. тексту, 34 рисунків, 2 таблиці, посилання на 19 літературних джерел та 6 конструкторських документів.

Ключові слова: нейронна мережа, датасет, десктоп-додаток, оптимізація, навчання.

Об'єктом розробки є додаток-конструктор для формування датасетів та навчання нейронних мереж.

Мета дипломного проєкту – розробка додатка-конструктора для формування датасетів та навчання нейронних мереж з метою спростити процес створення та навчання моделей машинного навчання.

Дипломний проєкт розглядає проблеми, пов'язані з формуванням наборів даних для навчання нейронних мереж. Основна увага приділяється вартості та складності збору та анутовання даних. В роботі проведено аналіз основних рішень та особливостей існуючих автоматизованих систем збору та анутовання даних, а також побудови та навчання нейронних мереж. Визначено переваги та недоліки цих систем.

Для здійснення дослідження було розроблено автоматизований desktop-додаток у середовищі PyCharm за допомогою мови програмування Python. Цей додаток надає користувачам зручність у створенні датасетів шляхом завантаження даних з відкритих ресурсів, редагування та розширення їх до необхідного обсягу, а також простоту в створенні та перевірці простих моделей нейронних мереж на власному комп'ютері. Desktop-додаток має достатній набір функцій для побудови датасетів та навчання нейронних мереж і надає зручне рішення у порівнянні з існуючими рішеннями.

Отримані результати можуть бути корисними при використанні їх в проєктах, які базуються на комп'ютерному зорі.

## SUMMARY

Pustovyi, D.D. Application-Builder for Dataset Formation and Neural Network Training. Igor Sikorsky KPI, Kyiv, 2023.

The project consists of 66 pages of text, 34 figures, 2 tables, references to 19 literary sources and 6 design documents.

Key words: neural network, dataset, desktop application, optimization, training.

The object of development is an application constructor for dataset generation and neural network training.

The aim of the diploma project is to develop an application constructor for dataset generation and neural network training to simplify the process of creating and training machine learning models.

The diploma project addresses the challenges associated with dataset generation for neural network training, with a focus on the cost and complexity of data collection and annotation. The work includes an analysis of existing automated systems for data collection, annotation, construction, and training of neural networks, identifying their advantages and limitations.

To conduct the research, an automated desktop application was developed using PyCharm IDE and the Python programming language. This application provides users with convenience in creating datasets by downloading data from open resources, editing and expanding them to the required volume, as well as ease in building and testing simple neural network models on their own computers. The desktop application offers a sufficient set of functions for dataset construction and neural network training, providing a more convenient solution compared to existing alternatives.

The obtained results can be useful in projects relying on computer vision applications.

Номер рядка	Формат	Позначення	Найменування	Кільк. аркушів	Номер екзем.	Примітка
1			<u>Документація загальна</u>			
2						
3			Знову розроблена			
4						
5	A4	IK91.018БАК.005 ПЗ	Пояснювальна записка	66		
6	A3	IK91.018БАК.005 Д1	Діаграма прецедентів	1		
7	A3	IK91.018БАК.005 Д2	Діаграма бізнеспроцесів	1		
8			BRMN			
9	A3	IK91.018БАК.005 Д3	Діаграма DFD	1		
10	A3	IK91.018БАК.005 Д4	Діаграма переходу інтерфейсів	1		
11			Ч.1			
12	A3	IK91.018БАК.005 Д5	Діаграма переходу інтерфейсів	1		
13			Ч.2			
14	A3	IK91.018БАК.005 Д6	Діаграма переходу інтерфейсів	1		
15			Ч.3			
16						
17						
18						
19						
20						
21						
22						
23						
24						
25						
26						
27						
28						

IK93.018БАК.005 ТП

Зм.	Аркуш	№ докум.	Підпис	Дата				
Розроб.		Пустовий Д.Д.			Застосунок для формування наборів даних та навчання нейронних мереж. Відомість проекту	Літ.	Аркуш	Аркушів
Керівн.		Коваль О.С.				Т	1	1
						КПІ ім. Ігоря Сікорського Група ІК-93		
Затв.								

**Пояснювальна записка  
до дипломного проєкту  
на тему: «Застосунок для формування наборів  
даних та навчання нейронних мереж»**

Київ – 2023 року

## ЗМІСТ

ПЕРЕЛІК СКОРОЧЕНЬ.....	4
ВСТУП.....	5
1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ ТА ПОСТАНОВКА ЗАДАЧІ .....	8
1.1 Загальні відомості про автоматизацію створення датасетів.....	8
1.2 Автоматизація та специфіка систем створення датасетів та навчання нейронних мереж.....	9
1.2.1 Особливості створення датасетів .....	9
1.2.2 Характеристика сервісів за призначенням .....	10
1.3 Аналіз існуючих систем .....	12
1.3.1 LabelImg .....	12
1.3.2 Веб-додаток Nyskel.....	14
1.3.3 Viso Suite .....	15
1.4 Постановка задачі.....	17
Висновки по розділу .....	18
2 МЕТОДИЧНЕ ТА ІНСТРУМЕНТАЛЬНЕ ЗАБЕЗПЕЧЕННЯ .....	20
2.1. Вимоги до додатку-конструктора.....	20
2.2. Загальні принципи проектування додатків .....	22
2.3. Підходи до розробки застосунків .....	23
2.3.1 Монолітний застосунок .....	23
2.3.2 Клієнт-серверна архітектура .....	24
2.4. Вибір технологій та платформи для реалізації додатка-конструктор .....	26
Висновки по розділу .....	26
3 ОПИС ПРОГРАМНОЇ РЕАЛІЗАЦІЇ .....	28
3.1 Архітектура програми.....	28
3.1.1 Створення датасету .....	29
3.1.2 Конструктор моделей.....	30

					<b>ІК93.018БАК.005 ПЗ</b>			
Зм.	Лист	№ докум.	Підпис		Додаток-конструктор для формування датасетів та навчання нейронних мереж. Пояснювальна записка	Літ.	Арк.	Аркушів
Розробив	Пустовий Д.Д			Т		2	66	
Перевірив	Коваль О.С.			КПІ ім. Ігоря Сікорського Група ІК-93				
Затв.								

3.1.3 Навчання і валідація моделі .....	31
3.2 Бібліотеки та фреймворки .....	32
3.2.1 TensorFlow .....	32
3.2.2 Keras.....	33
3.2.3 OIDv4 ToolKit .....	34
3.2.3.1 Open Image Dataset .....	34
3.2.4 PIL .....	35
3.3 Графічний інтерфейс користувача .....	36
3.3.1 PyQt.....	36
3.3.2 Qt Designer .....	37
3.4 Мова програмування.....	38
3.4.1 Python.....	38
3.4.2 PyCharm.....	40
3.5 Програмні рішення та розроблені алгоритми .....	41
3.5.1 Початкова сторінка .....	41
3.5.2 Структура проєкту користувача .....	44
3.5.3 Алгоритм збереження моделі.....	46
3.5.4 Алгоритми розширення датасету .....	48
Висновки по розділу .....	48
4 РОБОТА КОРИСТУВАЧА З ДОДАТКОМ.....	50
4.1 Інструкція користувача.....	50
Висновки по розділу .....	63
ВИСНОВКИ.....	64
ПЕРЕЛІК ПОСИЛАНЬ .....	65

## ПЕРЕЛІК СКОРОЧЕНЬ

НМ – нейронна мережа

ІК - Інтерфейс користувача

UI (User Interface) – Інтерфейс користувача

XML(eXtensible Markup Language) - Розширювана мова розмітки

API(Application Programming Interface) - Прикладний програмний інтерфейс

ФС - Файлова система

БД - База даних

PIL - Python Imaging Library

IDE(Integrated Development Environment) - Інтегроване середовище розробки

TF - TensorFlow

					ІК93.018БАК.005 ПЗ	Арк.
Зм.	Лист	№ докум.	Підпис	Дата		4

## ВСТУП

Штучний інтелект – це галузь комп'ютерних наук, яка займається розробкою комп'ютерних систем і програм, що здатні виконувати завдання, які зазвичай вимагають людського інтелекту. Ця технологія прагне створити комп'ютерні системи, здатні розуміти, мислити, вчитися, приймати рішення та взаємодіяти з людьми на природній мові.

Розвиток штучного інтелекту та глибокого навчання привів до значного зростання застосування нейронних мереж в різних сферах, включаючи комп'ютерне зорове сприйняття, розпізнавання образів та автоматичний аналіз даних. Однак, ефективне навчання нейронних мереж вимагає наявності якісних та добре анотованих наборів даних. Формування таких наборів даних є складним і часо та ресурсозатратним завданням, що потребує значних зусиль з боку дослідників.

Актуальність проблеми, яка зумовила вибір теми проекту, полягає в кількох факторах:

– зростання зацікавленості у штучному інтелекті: Останнім часом спостерігається значний розквіт досліджень та застосувань штучного інтелекту. Ця технологія стає все більш важливою у різних сферах, таких як медицина, транспорт, фінанси та багато інших;

– вартість та складність збору та анотування даних: Для успішного навчання нейронних мереж необхідні великі обсяги якісних та репрезентативних даних. Проте збір та анотування таких даних може бути трудомістким завданням, що вимагає великих зусиль, ресурсів і експертного вміння. Це ставить питання ефективності та доступності процесу формування датасетів;

– потреба в зручних інструментах для формування датасетів: Існуючі інструменти для формування датасетів можуть бути складними у використанні, неефективними або обмеженими у своїх можливостях. Розробка зручного та потужного додатка-конструктора, який дозволяє легко створювати, редагувати та розширювати датасети, стала актуальною задачею.

					ІК93.018БАК.005 ПЗ	Арк.
Зм.	Лист	№ докум.	Підпис	Дата		5

Таким чином, проблеми, пов'язані з формуванням датасетів для навчання нейронних мереж, виникли внаслідок широкого застосування штучного інтелекту і потреби у зручних та ефективних інструментах для цього процесу. Розробка автоматизованого додатка-конструктора є важливим кроком у полегшенні процесу формування датасетів та покращенні навчання нейронних мереж.

Дослідження спрямоване на розробку та вдосконалення цього додатка з метою спрощення процесу формування датасетів і навчання нейронних мереж. Об'єкт дослідження включає такі аспекти:

– формування датасетів: Дослідження орієнтоване на створення механізмів, які дозволяють користувачу з легкістю створювати датасети. Це може включати завантаження даних з відкритих ресурсів, можливість редагування та розширення даних до необхідного обсягу;

– навчання нейронних мереж: Дослідження спрямоване на розробку інструментів, які дозволяють користувачу створювати та навчати прості моделі нейронних мереж. Це може включати підтримку різних типів нейронних мереж, параметризацію моделей та можливість перевірки роботи моделі на власному комп'ютері.

Загальна мета нашого дослідження полягає в розробці ефективного додатка-конструктора, який допоможе дослідникам і практикам спростити процес формування датасетів та навчання нейронних мереж. Ми прагнемо забезпечити зручність, швидкість та інтуїтивну зрозумілість у використанні додатку, щоб користувачі могли легко створювати високоякісні датасети та навчати потужні нейронні мережі.

Для досягнення цієї мети ми визначили такі завдання:

– проаналізувати проблеми та виклики, пов'язані з формуванням наборів даних для навчання нейронних мереж;

– вивчити основні підходи та методи, що використовуються для створення датасетів та навчання нейронних мереж;

– розробити додаток-конструктор, який надасть зручність та ефективність у формуванні датасетів та навчанні нейронних мереж;

					ІК93.018БАК.005 ПЗ	Арк.
Зм.	Лист	№ докум.	Підпис	Дата		6

– перевірити ефективність та продуктивність розробленого додатка на реальних даних та задачах навчання нейронних мереж.

Дипломний проєкт складається з наступних розділів: вступ, огляд літератури, аналіз предметної області та постановка задачі, методичне та інструментальне забезпечення, опис програмної реалізації, робота користувача з додатком, висновки та додатки. Кожен розділ включає необхідні теоретичні та практичні аспекти, які допоможуть досягти поставлених цілей дослідження.

Цей дипломний проєкт має практичне значення для дослідників, студентів та спеціалістів, які займаються розробкою та навчанням нейронних мереж. Наш додаток-конструктор може сприяти полегшенню процесу формування датасетів та навчання нейронних мереж, що в свою чергу сприятиме розвитку і застосуванню штучного інтелекту та комп'ютерного зору у різних галузях.

					ІК93.018БАК.005 ПЗ	Арк.
Зм.	Лист	№ докум.	Підпис	Дата		7

# 1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ ТА ПОСТАНОВКА ЗАДАЧІ

## 1.1 Загальні відомості про автоматизацію створення датасетів

Створення датасетів є ключовим етапом перед навчанням нейронних мереж. Добре підготовлений та репрезентативний датасет є важливою передумовою для успішного навчання моделей. Від якості та репрезентативності датасету залежить точність та ефективність навчання моделей. Про те які можуть виникнути проблеми при не коректному датасеті описано автором Adrian Rosebrock в книзі “Deep Learning for Computer Vision with Python”. Автор зазначає, що неминуче під час тренування глибокої нейромережі на тренувальному наборі та його оцінки на тестовому наборі (з отриманням високої точності), виникає проблема, коли застосовують навчену модель до зображень, що не належать до жодного з цих наборів. Виявляється, що нейромережа показує незадовільні результати [1].

Автоматизація створення датасетів передбачає використання комп'ютерних алгоритмів та програм для збору, обробки, анотування та підготовки даних, які будуть використовуватись для навчання нейронних мереж. Це включає в себе завантаження даних з різних джерел, перетворення форматів даних, фільтрацію та очищення даних, а також маркування та анотування даних з метою надання їм семантичного значення. Автоматизація цих процесів дозволяє прискорити та полегшити роботу з великими обсягами даних, забезпечуючи їх якість та консистентність.

Побудова моделей нейронних мереж також може бути автоматизованою за допомогою спеціальних алгоритмів та інструментів. Це включає в себе вибір та конфігурацію архітектури нейронної мережі, встановлення оптимальних значень гіперпараметрів, навчання моделі на вхідних даних, валідацію та оцінку результатів. Автоматизація побудови моделей дозволяє зменшити суб'єктивність та помилки, а також забезпечити більшу ефективність та репрезентативність моделей.

					ІК93.018БАК.005 ПЗ	Арк.
Зм.	Лист	№ докум.	Підпис	Дата		8

## 1.2 Автоматизація та специфіка систем створення датасетів та навчання нейронних мереж

### 1.2.1 Особливості створення датасетів та навчання нейронних мереж

Особливості створення датасетів та навчання нейронних мереж включають ряд важливих аспектів, що впливають на ефективність та якість моделей машинного навчання. Ось деякі особливості, які варто враховувати:

**Розмір датасету:** Для успішного навчання нейронних мереж необхідно мати достатньо великий датасет, що включає достатню кількість зразків даних. Великі датасети дозволяють моделям краще узагальнювати та знаходити загальні закономірності.

**Розподіл класів:** Якщо датасет містить класи з нерівномірним розподілом, наприклад, один клас має значно менше зразків, ніж інші, це може призвести до небалансу та погіршити результати навчання моделі. Необхідно забезпечити збалансованість класів у датасеті або застосувати відповідні методи боротьби з нерівноважністю класів.

**Аугментація даних:** Аугментація даних – це процес штучного збільшення розміру датасету шляхом застосування різних перетворень до існуючих зразків даних. Це може включати зміну масштабу, повороти, зміщення, розмивання тощо. Аугментація даних сприяє розширенню різноманітності даних та покращує роботу моделі при зустрічі з новими зображеннями. Ці принципи були описані в “Hands-On Image Processing with Python” за авторством Sandipan Dey [2].

**Якість даних:** Якість даних є критичним фактором у навчанні нейронних мереж. Некоректні або забруднені дані можуть вплинути на точність та надійність моделі. Тому важливо виконувати контроль якості даних, видаляти шум, аномалії та некоректні зразки даних.

**Навчання та валідація:** Датасет зазвичай розділяється на навчальну та валідаційну вибірки. Навчальна вибірка використовується для тренування моделі, а валідаційна – для оцінки її ефективності та підбору параметрів. Важливо

					ІК93.018БАК.005 ПЗ	Арк.
Зм.	Лист	№ докум.	Підпис	Дата		9

забезпечити репрезентативність обох вибірок, щоб модель змогла добре узагальнювати.

Перевірка та тестування: Після навчання моделі на навчальній вибірці необхідно перевірити її ефективність на тестовій вибірці, яка раніше не була використана в процесі навчання. Це дозволяє оцінити загальну продуктивність моделі та її здатність до узагальнення на нові дані.

### 1.2.2 Характеристика сервісів за призначенням

Датасет – це сукупність даних, використовуваних для навчання та валідації моделей машинного навчання. Він може включати різноманітні типи даних, такі як текст, зображення, аудіо та числові значення. Датасет є фундаментальною основою для навчання нейронних мереж, оскільки надає моделі доступ до різноманітних прикладів та вхідних параметрів. Важливі аспекти створення датасету включають збір та анотування даних, їхню нормалізацію та підготовку. Якісний та репрезентативний датасет допомагає моделям машинного навчання вивчати закономірності та робити точні передбачення на нових невідомих даних. Датасети грають ключову роль у розвитку та застосуванні штучного інтелекту та машинного навчання.

Модель нейронної мережі – це математична конструкція, яка включає штучні нейрони та ваги. Вона призначена для вирішення конкретної задачі шляхом обробки вхідних даних. Модель отримує дані з датасету та застосовує внутрішню структуру та ваги, щоб згенерувати відповіді або зробити передбачення. Штучні нейрони в моделі працюють схожим чином на біологічні нейрони, сполучаючись і передаючи сигнали один одному. Ваги моделі встановлюються шляхом навчання на датасеті, що дозволяє моделі "вчитися" і покращувати свої результати з часом. Моделі нейронних мереж використовуються для широкого спектру завдань, включаючи класифікацію, регресію, обробку зображень та мови, рекомендації та багато іншого.

					ІК93.018БАК.005 ПЗ	Арк.
Зм.	Лист	№ докум.	Підпис	Дата		10

Препроцесинг даних є важливим етапом перед навчанням моделі і передбаченням. Це процес підготовки даних для подальшого використання, що включає ряд операцій. Нормалізація даних стандартизує їх значення, забезпечуючи однаковий масштаб. Перетворення даних може включати зміну формату або кодування, щоб забезпечити сумісність з моделлю. Обрізання даних може відбуватися для зменшення обсягу або видалення непотрібних даних. Анотування даних означає присвоєння міток або класифікацію елементів датасету. Ці операції допомагають забезпечити якість та правильне

представлення даних, що сприяє ефективному навчанню моделі та досягненню точних результатів.

Тренування моделі – це критичний етап у процесі машинного навчання, який включає навчання моделі на підготовленому датасеті. Під час тренування, модель аналізує вхідні дані та їх мітки, і поступово відпрацьовує свої ваги, зусиллями алгоритму оптимізації, щоб максимізувати точність передбачення. Цей процес може вимагати значних обчислювальних ресурсів, включаючи могутні графічні процесори та паралельні обчислення, а також займати тривалий час, особливо при великих обсягах даних та складних моделях. Тренування моделі є ітеративним процесом, де модель поступово вдосконалюється шляхом аналізу помилок та корекції своїх ваг. Це важливий крок для досягнення високої точності та ефективності моделі під час подальшого застосування.

Оцінка та валідація моделі є важливим етапом у процесі машинного навчання, що дозволяє оцінити її ефективність та здатність до узагальнення. Цей процес включає використання незалежного тестового датасету для оцінки моделі на нових, раніше невидених даних. Під час валідації, використовуються різні метрики оцінки, такі як точність, відновлення, F-мера та інші, щоб кількісно виміряти якість передбачень моделі. Це дозволяє оцінити, наскільки добре модель працює та як вона здатна узагальнювати на нові дані. Оцінка та валідація моделі є важливими кроками для забезпечення її надійності та використання в практичних задачах.

					ІК93.018БАК.005 ПЗ	Арк.
Зм.	Лист	№ докум.	Підпис	Дата		11

Інтерфейс користувача – це важливий компонент додатку, який забезпечує зручну взаємодію користувача з системою. Цей компонент надає широкі можливості, такі як завантаження та перегляд датасетів, налаштування параметрів моделі, відображення результатів та багато іншого. ІК забезпечує інтуїтивний та простий спосіб взаємодії, що дозволяє користувачам ефективно використовувати систему без необхідності глибокого розуміння технічних деталей. Це дозволяє навіть користувачам без попереднього досвіду з машинним навчанням використовувати та налаштовувати моделі для своїх конкретних потреб. ІК сприяє покращенню взаємодії та зручності використання системи, що важливо для успішного впровадження та прийняття додатку користувачами.

### 1.3 Аналіз існуючих систем

#### 1.3.1 LabelImg

LabelImg – це інструмент з відкритим вихідним кодом, який надає можливість маркування зображень за допомогою графічного інтерфейсу (рисунок 1.1) [3]. Розроблений у 2015 році, цей програмний продукт на мові Python використовує потужність QT для створення зручного та простого UI. Це незамінний інструмент для створення датасетів у галузі комп'ютерного зору, де можна маркувати кілька сотень зображень. Завдяки можливості збереження анотацій у форматі XML (рисунок 1.2), відповідному стандарту PASCAL VOC, LabelImg робить процес маркування простим та зручним. Зауважимо, що цей формат анотацій може не підходити для всіх моделей виявлення об'єктів, але залишається широко використовуваним у галузі ImageNet

Цей додаток дозволяє обробляти не лише окремі зображення, але й вибирати цілі папки з датасетом, а за допомогою кнопок "Вперед" і "Назад" зручно переміщатися по них. Особливо корисною є можливість зберігання всіх внесених змін і побачення результатів своєї роботи, повернувшись до попереднього зображення після обробки. Цей додаток відрізняється своєю простотою використання, але, як будь-якому продукту, йому також притаманні недоліки.

					ІК93.018БАК.005 ПЗ	Арк.
Зм.	Лист	№ докум.	Підпис	Дата		12

Головним об'єктивним недоліком цього додатку є його обмежений функціонал, який обмежується можливістю лише створювати рамки для виділення конкретного об'єкта на зображеннях, з метою пізнішого використання цих рамок для навчання нейронних мереж. З моєї точки зору, ще одним недоліком є відсутність підтримки всіх етапів життєвого циклу систем штучного інтелекту, таких як збір даних, анотування, навчання, розробка, розгортання та моніторинг систем глибокого навчання. Ці обмеження можуть ускладнити розробку та використання комплексних систем штучного інтелекту.

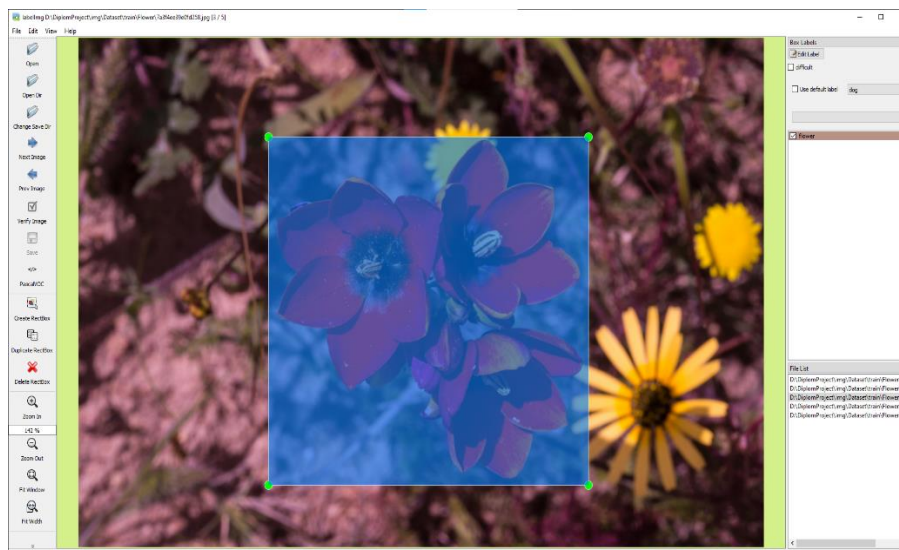


Рисунок 1.1 – Інтерфейс додатку LabelImg

```

1 <annotation verified="yes">
2   <folder>Flower</folder>
3   <filename>7a3f4ee39e0fd258.jpg</filename>
4   <path>D:\DiplomProject\img\Dataset\train\Flower\7a3f4ee39e0fd258.jpg</path>
5   <source>
6     <database>Unknown</database>
7   </source>
8   <size>
9     <width>1024</width>
10    <height>683</height>
11    <depth>3</depth>
12  </size>
13  <segmented>0</segmented>
14  <object>
15    <name>flower</name>
16    <pose>Unspecified</pose>
17    <truncated>0</truncated>
18    <difficult>0</difficult>
19    <bndbox>
20      <xmin>284</xmin>
21      <ymin>133</ymin>
22      <xmax>762</xmax>
23      <ymax>600</ymax>
24    </bndbox>
25  </object>
26 </annotation>

```

Рисунок 1.2 – Результат роботи додатку LabelImg

					ІК93.018БАК.005 ПЗ	Арк.
Зм.	Лист	№ докум.	Підпис	Дата		13

### 1.3.2 Веб-додаток Nyskel

Для створення комплексного додатку, ми провели дослідження різних рішень, які надають можливості для навчання та валідації НМ.

Nyskel – це інноваційна платформа, що розробляє швидкий, потужний та ергономічний API для налаштованого машинного навчання [4]. Вона створена з метою спростити процес використання машинного навчання для розробників, усуваючи необхідність у власній команді машинного навчання. Завдяки Nyskel, розробники можуть легко і швидко інтегрувати передові методи машинного навчання у свої додатки всього за кілька хвилин. Це відкриває безмежні можливості для покращення функціональності та ефективності додатків, дозволяючи використовувати передові алгоритми навчання для розв'язання складних завдань.

Використовуючи UI, користувач може легко створити нову модель своєї нейронної мережі. Він може імпортувати свій власний датасет та надати кожному типу даних свій власний ярлик, який використовується для класифікації (рисунок 1.3). Після імпорту всього датасету та надання ярликів, починається автоматичний процес навчання. Після завершення навчання можна перевірити навчену модель, яка поверне клас, до якого належить кожне зображення, разом з процентом точності (рисунок 1.4). Таким чином, користувач отримує зручний спосіб створення та перевірки власних моделей нейронних мереж для класифікації зображень.

Перевагами цього веб-додатку є можливість навчання нейронних мереж, підтримка різних типів вхідних даних, таких як зображення, текст або таблиці, а також декілька вихідних результатів, таких як класифікація, тегування і пошук. Крім того, є можливість отримати API для кожного етапу навчання нейронної мережі.

Серед недоліків можна виділити декілька пунктів. По-перше, необхідно мати готовий набір даних для навчання. По-друге, не можна вибрати конкретну модель для навчання, наприклад, пряме поширення або рекурентну мережу.

					ІК93.018БАК.005 ПЗ	Арк.
Зм.	Лист	№ докум.	Підпис	Дата		14

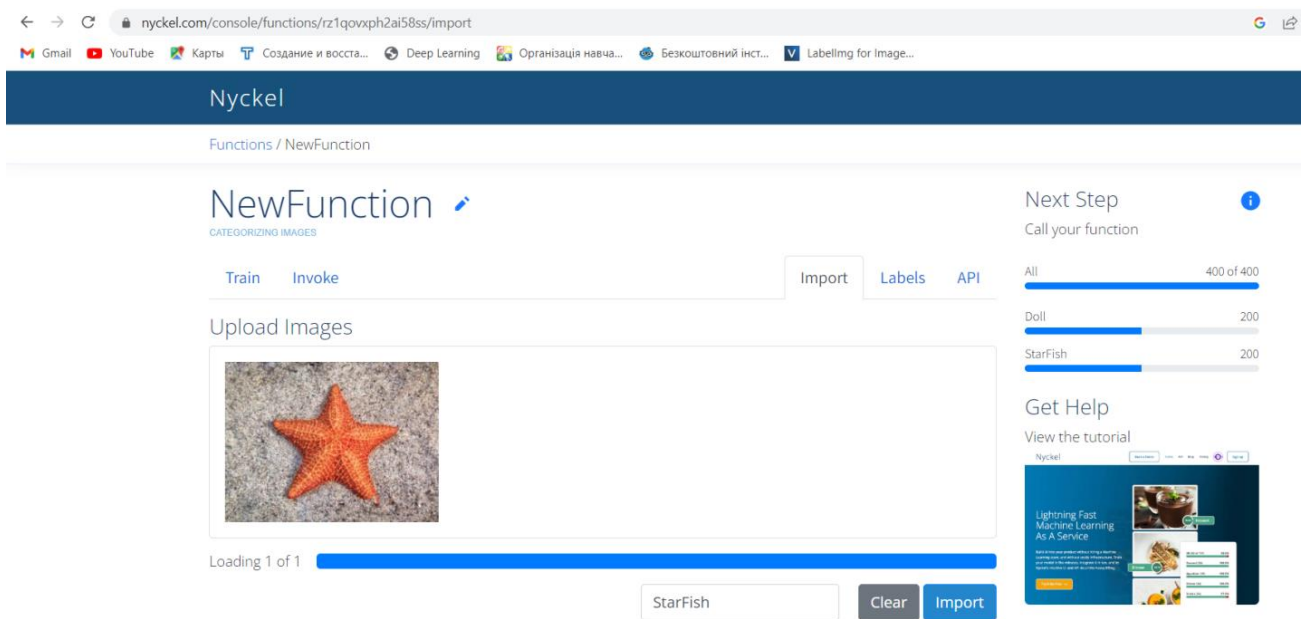


Рисунок 1.3 – Інтерфейс імпортування датасету

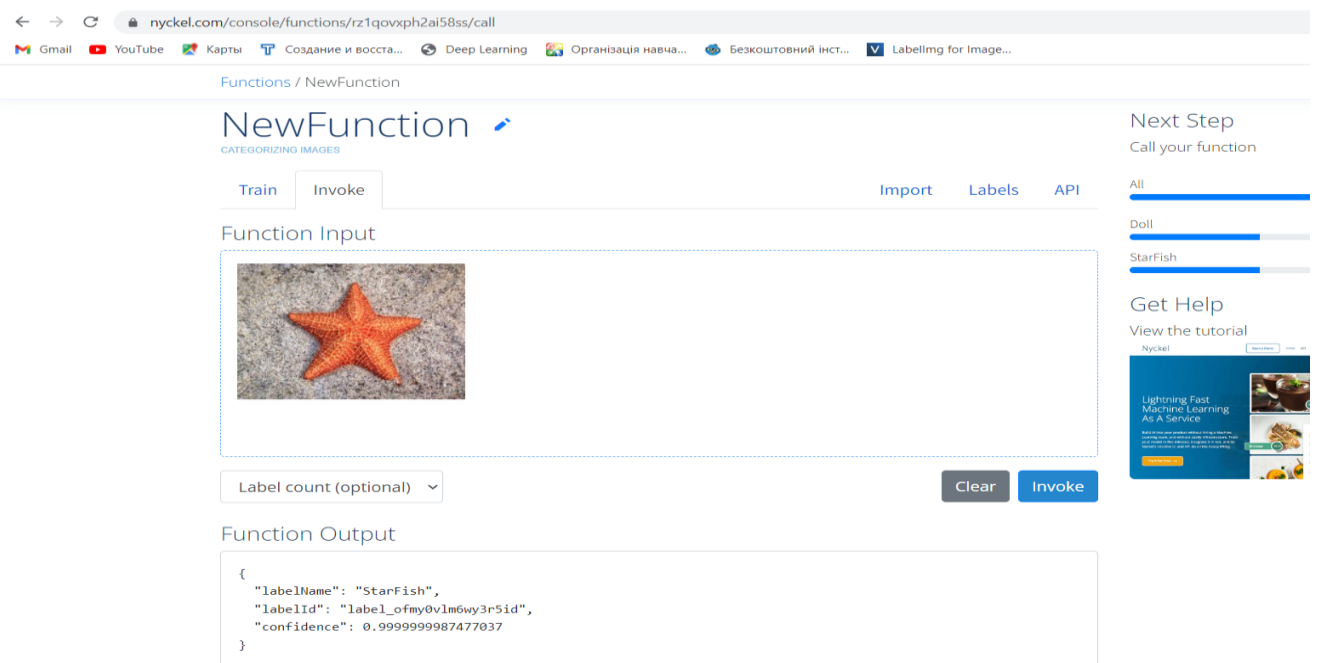


Рисунок 1.4 – Інтерфейс валідації НМ

### 1.3.3 Viso Suite

Viso Suite – це інноваційна платформа розробки додатків, яка надає широкий набір інструментів та функціональностей для швидкої та ефективної розробки систем комп'ютерного зору в корпоративному середовищі [5]. Завдяки

					ІК93.018БАК.005 ПЗ	Арк.
Зм.	Лист	№ докум.	Підпис	Дата		15

безкодовому підходу, розробники можуть автоматизувати ручну роботу та знизити витрати на розробку. Відповідаючи вимогам масштабованості, конфіденційності та безпеки, Viso Suite забезпечує надійний та захищений процес роботи на кожному етапі проєкту. Незалежно від потреб вашої організації, Viso Suite забезпечує все необхідне для успішної розробки та впровадження систем комп'ютерного зору. Основні функції Viso Suite:

– збір даних для анотування комп'ютерного зору відбувається за допомогою Viso Suite, яка надає вбудовані автоматизовані можливості збору тренувальних даних (рисунок 1.5);

– анотування зображень та відеоданих (рисунок 1.6) ;

– навчайте та керуйте моделями штучного інтелекту. Керування своїми моделями штучного інтелекту у всіх фреймворках у бібліотеці моделей. Можливість вибирати з готових моделей, імпортувати або навчати свої власні моделі штучного інтелекту (рисунок 1.7).

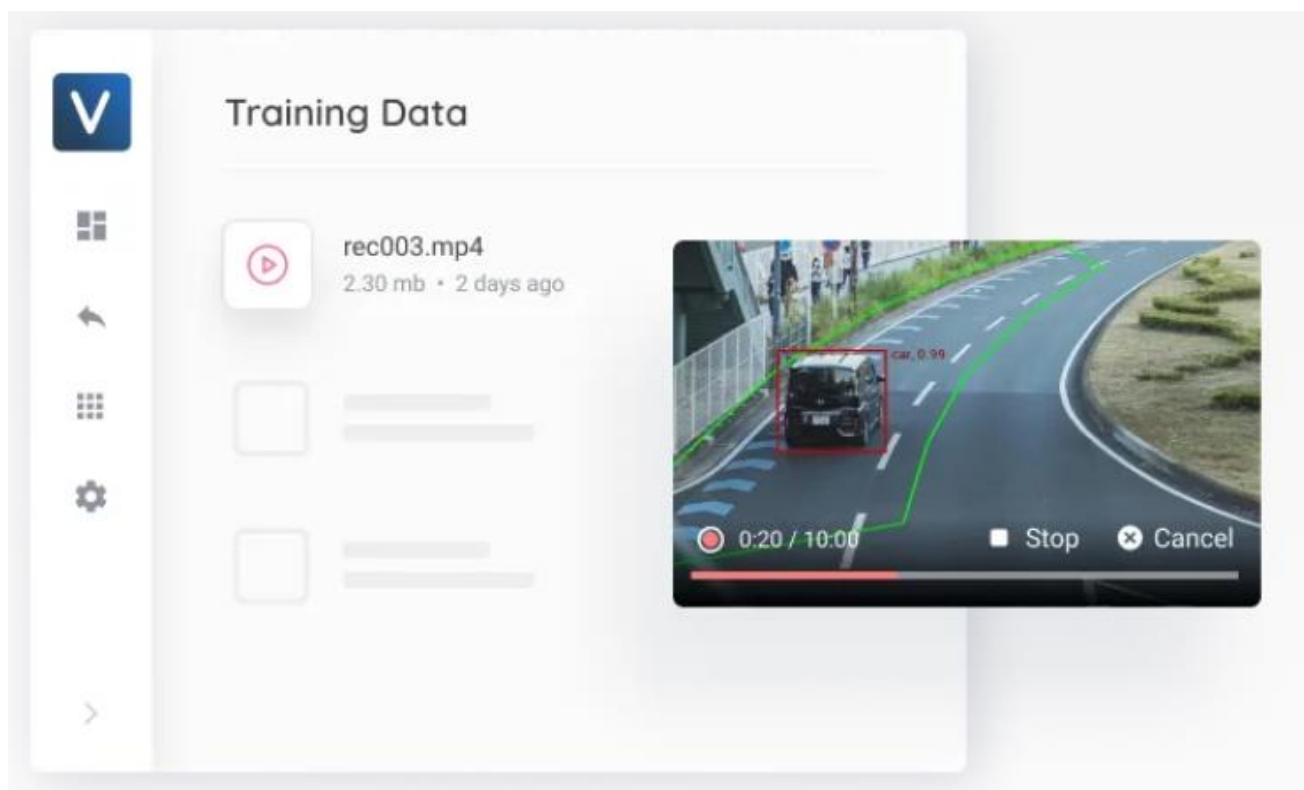


Рисунок 1.5 – Інтерфейс збору даних

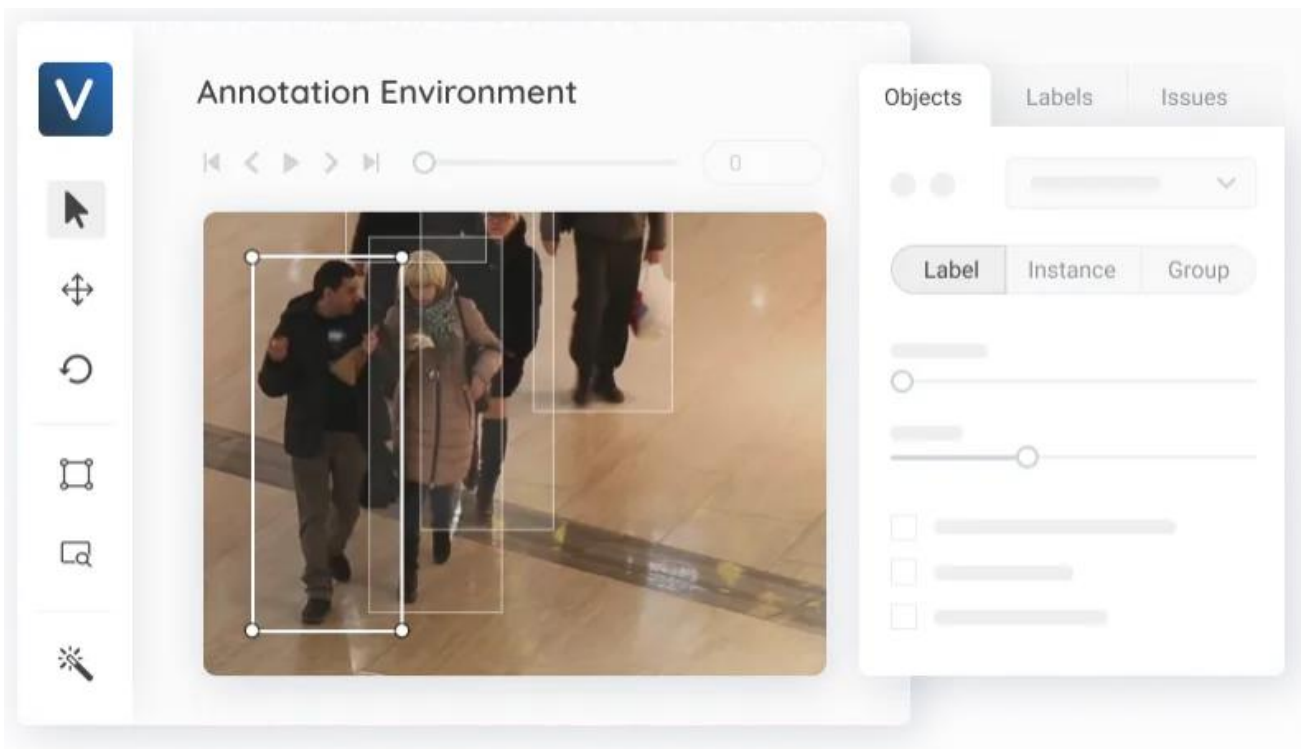


Рисунок 1.6 – Інтерфейс анотування

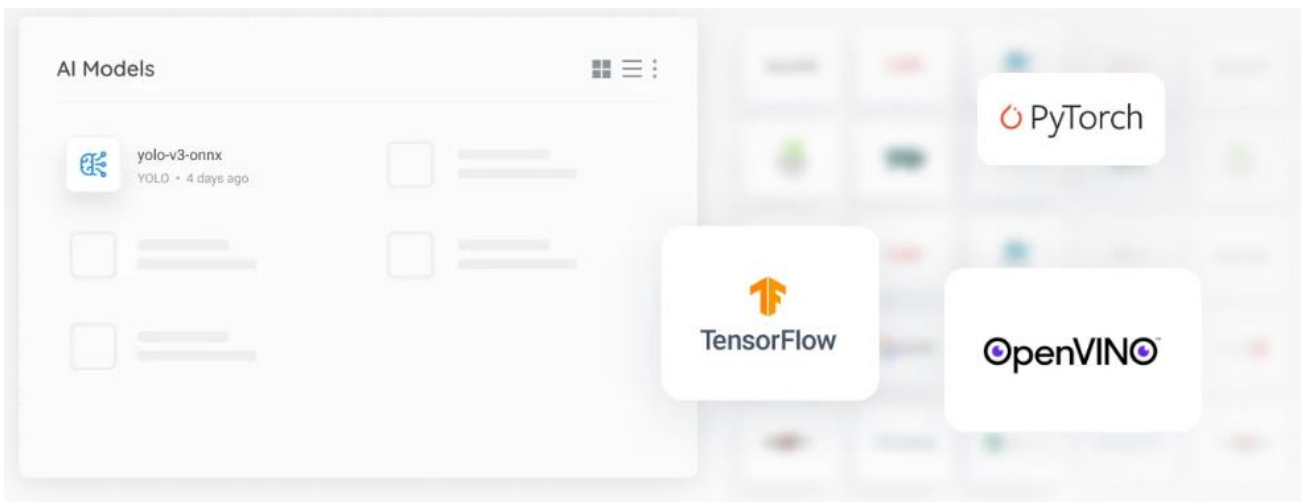


Рисунок 1.7 – Інтерфейс створення моделі

#### 1.4 Постановка задачі

Після аналізування наявних рішень, стає очевидним, що проблема неповноти та недостатньої відповідності базовим вимогам існує, що ускладнює процес створення повного датасету.

Отже основною метою дипломного проекту є розробка власної автоматизованої системи, яка дозволить користувачам легко створювати готовий до навчання датасет та перевіряти його на створених НМ. З цією метою було розроблено систему з назвою "Constructor for Dataset Forming and Training NN", яка поєднує основні переваги існуючих рішень та відповідає наступним вимогам:

- зручний інтерфейс користувача для легкого використання системи;
- автоматизоване формування датасету з можливістю розширення до потрібної кількості;
- можливість вибору різних алгоритмів машинного навчання та нейромереж;
- наявність інструментів для валідації та оцінки точності навчених моделей;
- можливість експорту результатів;
- надійність та стабільність системи під час навчання та використання нейромереж;
- підтримка розширення системи для включення нових функцій та алгоритмів;
- можливість копіювання даних та моделей;
- вбудована документація та підтримка для користувачів.

Предметом дослідження в даній роботі виступають методи роботи з картинками для формування датасетів та способи створення та навчання різних моделей НМ за умови використання десктоп-додатку, а об'єктом дослідження є формування датасетів та навчання НМ.

Необхідністю роботи зі створення даного додатку є задоволення потреб користувачів у швидкому і зручному формуванні датасетів та можливості експортувати результати.

### Висновки по розділу

У даному розділі досліджено особливості формування датасетів і навчання штучного інтелекту, проаналізовано існуючі рішення з цих галузей, їх переваги та недоліки. Всі практичні навички було набуто в таких джерелах, як книга “ Python

					ІК93.018БАК.005 ПЗ	Арк.
Зм.	Лист	№ докум.	Підпис	Дата		18

Deep Learning Cookbook ” від Indra den Bakker та інтернет курс “ Python for Computer Vision with OpenCV and Deep Learning” на платформі PyImageSearch [6-7]. Метою розроблюваного додатку є автоматизація формування датасетів і забезпечення їх перевірки на побудованій моделі штучного інтелекту.

Було визначено, що постає проблема недостатньої ефективності та складнощів у процесі формування датасетів та навчання нейронних мереж. Існуючі рішення в цій сфері не повністю задовольняють базові вимоги, що призводить до затримок у роботі, неповних та неконсистентних наборів даних, а також складнощів у перевірці та оцінці результатів. Така проблема має вплив на швидкість розробки та впровадження нейронних мереж, а також на якість та точність отриманих моделей. Також було сформульовано головну задачу дипломного проєкта й завдання, що необхідно виконати для її реалізації.

Було складено вимоги до розробки системи та безпосередньо основні пункти плану її розробки. Отже, треба розробити систему за наступною діаграмою (рисунок 1.8):

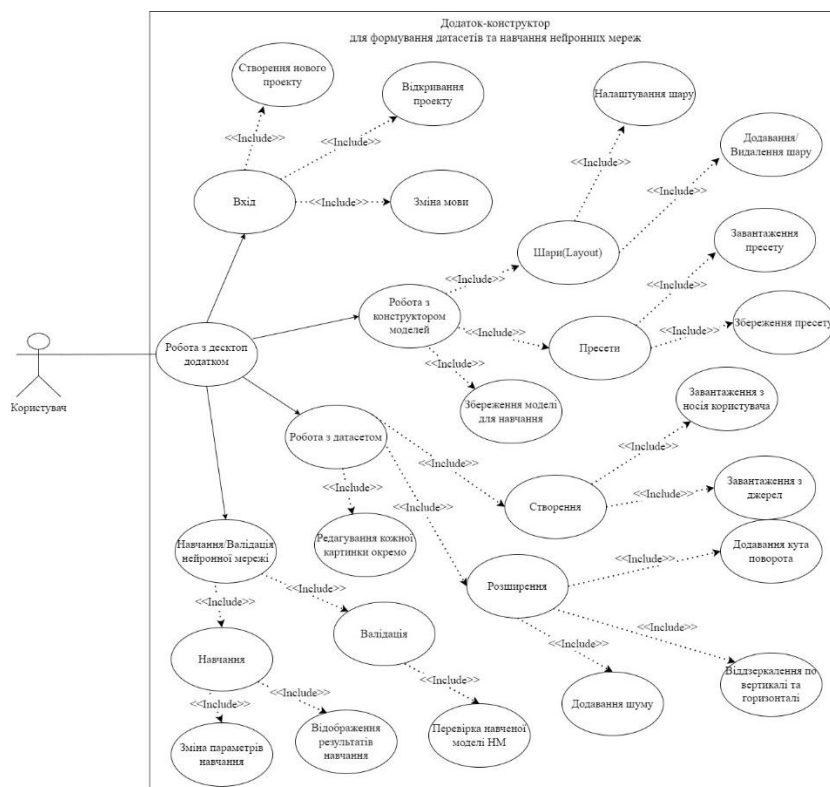


Рисунок 1.8 – Діаграма прецедентів

Зм.	Лист	№ докум.	Підпис	Дата

## 2 МЕТОДИЧНЕ ТА ІНСТРУМЕНТАЛЬНЕ ЗАБЕЗПЕЧЕННЯ

### 2.1 Вимоги до додатку-конструктора

У сфері машинного навчання та штучного інтелекту, якісні датасети є необхідною складовою для тренування та ефективного функціонування нейронних мереж. Однак, існують ряд проблем, з якими стикаються дослідники та розробники у процесі формування датасетів.

По-перше, нестача якісних датасетів є серйозним викликом. Багато існуючих наборів даних можуть бути неповними або не відповідати конкретним завданням, з якими працюють дослідники. Відсутність належних даних може суттєво ускладнити процес тренування нейронних мереж та призвести до недостатньо точних результатів.

По-друге, проблема забрудненості та неповноти даних виникає при зборі існуючих датасетів. Це означає, що дані можуть містити помилки, неточності, аномалії або бути неповними. Наявність таких даних може негативно вплинути на процес навчання нейронних мереж та призвести до некоректних або неточних прогнозів.

По-третє, вартість та складність збору та анування даних також становлять проблему. Залежно від завдання та області дослідження, збір достатньо великого та репрезентативного набору даних може вимагати значних ресурсів, які можуть бути обмеженими для багатьох дослідників та розробників. Крім того, анування даних, тобто процес маркування, класифікації або розпізнавання об'єктів на зображеннях, може бути часо та працезатратним завданням.

Враховуючи ці проблеми, розробка додатка-конструктора для формування датасетів та навчання нейронних мереж стає важливим завданням. Цей додаток покликаний вирішити зазначені проблеми шляхом надання користувачам інструментів, які спростять та оптимізують процес формування якісних датасетів.

Перш за все, додаток-конструктор надає можливість користувачам самостійно створювати датасети за своїми потребами та вимогами. Він дозволяє встановлювати різні параметри та налаштування, такі як розмір датасету, розподіл

					ІК93.018БАК.005 ПЗ	Арк.
Зм.	Лист	№ докум.	Підпис	Дата		20

класів, різноманітність даних та інші. Це дає можливість вирішити проблему нестачі якісних датасетів шляхом створення нових наборів даних, які відповідають конкретним завданням.

Друга проблема – забрудненість та неповнота даних, також вирішується за допомогою додатка-конструктора. Він надає користувачам зручний інтерфейс для анування даних, де можна маркувати об'єкти, класифікувати дані, визначати регіони та здійснювати інші операції. Це дозволяє ефективно очищати та підготовляти дані, усуваючи помилки та неточності, що сприяє покращенню якості навчання нейронних мереж.

Крім того, додаток-конструктор зменшує вартість та складність збору та анування даних шляхом автоматизації певних процесів. Наприклад, він може використовувати методи комп'ютерного зору для автоматичного розпізнавання об'єктів на зображеннях або застосовувати попередні аотації для подальшого в навчання. Це дозволяє значно зекономити час та зусилля, які розробники та дослідники витрачають на ручне анування великих обсягів даних.

Додаток-конструктор також може надавати інструменти для перевірки та валідації даних, що дозволяє виявляти потенційні проблеми або аномалії в датасеті. Це сприяє поліпшенню якості та надійності навчання нейронних мереж, оскільки виключає можливість використання некоректних або неповних даних.

Застосування додатка-конструктора для формування датасетів та навчання нейронних мереж має потенціал значно спростити та оптимізувати процес розробки та впровадження моделей штучного інтелекту. Він дозволяє дослідникам та розробникам зосередитись на самому алгоритмі та вдосконаленні моделі, мінімізуючи проблеми, пов'язані з якістю та доступністю даних.

Загалом, розробка додатка-конструктора для формування датасетів та навчання нейронних мереж вирішує ключові проблеми у сфері машинного навчання, такі як нестача якісних датасетів, забрудненість та неповнота даних, а також вартість та складність збору та анування даних. Цей інструмент дозволяє користувачам ефективно формувати власні датасети, очищати та підготовляти дані та прискорювати процес навчання нейронних мереж.

					ІК93.018БАК.005 ПЗ	Арк.
Зм.	Лист	№ докум.	Підпис	Дата		21

## 2.2 Загальні принципи проектування додатків

Було вирішено розробити програму «Constructor for Dataset Forming and Training NN» у вигляді десктоп-застосунку. Основною причиною цього є те, що не потрібно виділяти ресурси для навчання НМ та формування датасету, вони будуть братися з системи користувача.

Загальні принципи проектування додатків включають набір основних принципів, які слід враховувати під час розробки будь-якого додатка, включаючи застосунок для формування наборів даних та навчання нейронних мереж. Основні принципи проектування додатків включають:

- модульність: Розділення функцій та компонентів додатка на окремі модулі або компоненти, що сприяє легкості розробки, тестування та підтримки;
- розширюваність: Забезпечення можливості додавання нових функцій і розширення функціональності додатка без необхідності внесення значних змін у вже існуючий код;
- інтуїтивний інтерфейс: Розробка зручного та легкозрозумілого інтерфейсу користувача, який дозволяє легко взаємодіяти з додатком і виконувати необхідні завдання;
- ефективність: Мінімізація часу відгуку додатка, оптимізація роботи алгоритмів та використання ресурсів для забезпечення швидкої та ефективної роботи;
- надійність: Розробка додатка, який працює стабільно та надійно, враховуючи можливі ситуації виникнення помилок та відновлення після непередбачених ситуацій;
- безпека: Захист даних користувачів та забезпечення безпеки обробки та збереження інформації, що використовується в додатку;
- переносимість: Забезпечення можливості запуску додатка на різних платформах та операційних системах.

					ІК93.018БАК.005 ПЗ	Арк.
Зм.	Лист	№ докум.	Підпис	Дата		22

## 2.3 Підходи до розробки застосунків

Перше питання, яке постає перед розробником, який буде проектувати десктоп-застосунок: який підхід обрати до створення десктоп-застосунку та яким чином платформа, де буде розгортатися застосунок повинна вплинути на його архітектуру.

При розробці застосунків існують різні архітектурні підходи, які дозволяють організувати структуру, взаємодію та функціональність програмного продукту. Ось кілька архітектурних підходів, які можуть бути використані при розробці додатка-конструктора:

- монолітна архітектура;
- клієнт-серверна архітектура;
- пайпи і фільтри (Pipes and filters);
- plugin;
- front end та back end.

Далі буде більш детально розписано саме про ті підходи, які ми розглядали для розробки додатку.

### 2.3.1 Монолітний застосунок

Монолітна архітектура відноситься до програмного дизайну, де вся логіка і функціональність програми знаходяться в одному проєкті без розділення на окремі компоненти чи модулі. Такий проєкт компілюється в єдину збірку і розгортається як єдине ціле.

Наприклад, стандартний Python проєкт може бути прикладом монолітної архітектури, де весь код, пов'язаний з представленням, бізнес-логікою та доступом до даних, знаходиться в одному проєкті (рисунок 2.1). Зазвичай, такі проєкти можуть використовувати структуру з окремими папками для розділення функціональних частин, але це не означає фізичного розділення на окремі компоненти.

					ІК93.018БАК.005 ПЗ	Арк.
Зм.	Лист	№ докум.	Підпис	Дата		23

Монолітний підхід є простим і зручним для розробки, розуміння структури та розгортання проєкту. Він особливо підходить для невеликих проєктів. Однак, цей підхід може стати обмежуючим у великих та складних проєктах, де може бути важко підтримувати та масштабувати програму без розділення на окремі компоненти або сервіси. Sam Newman детально описав недоліки монолітної архітектури порівняно з мікросервісною архітектурою. Він не тільки показав, де монолітна архітектура виявляється менш ефективною, але й надав рекомендації про те, в яких випадках вона може бути кращим варіантом [8].

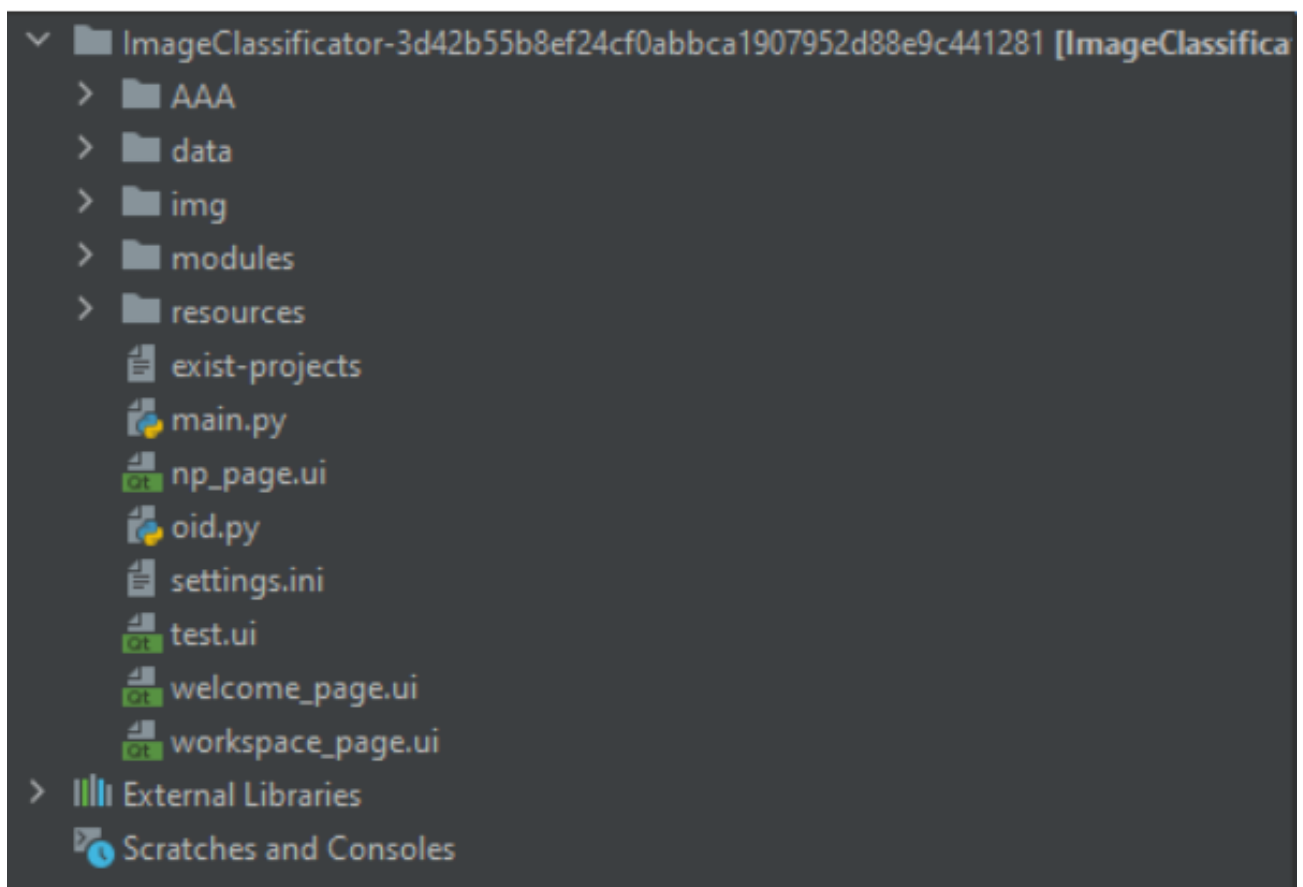


Рисунок 2.1 – Приклад розділення монолітного проєкту

### 2.3.2 Клієнт-серверна архітектура

Клієнт-серверна архітектура є одним з найпоширеніших підходів у розробці десктоп-застосунків. У цій архітектурі програма розбивається на дві основні частини: клієнтську (client) та серверну (server) (рисунок 2.2).

					ІК93.018БАК.005 ПЗ	Арк.
Зм.	Лист	№ докум.	Підпис	Дата		24

Клієнтська частина відповідає за взаємодію з користувачем та представленням даних. Вона може бути реалізована у вигляді десктопного додатку, який встановлюється на комп'ютер користувача, або у вигляді веб-додатку, який відкривається у браузері. Клієнтська частина відправляє запити до сервера і отримує від нього необхідні дані або результати обробки.

Серверна частина відповідає за обробку запитів, збереження та управління даними, бізнес-логіку та виконання ресурсоємних операцій. Вона може бути реалізована як окремий сервер або веб-сервер, до якого клієнти надсилають свої запити. Серверна частина відповідає за взаємодію з базою даних, забезпечення безпеки та обробку бізнес-правил.

Переваги клієнт-серверної архітектури полягають у розділенні обов'язків між клієнтом та сервером, що спрощує розробку та підтримку додатка. Вона дозволяє централізувати управління даними та забезпечити консистентність інформації. Крім того, клієнт-серверна архітектура підтримує можливість розширення та масштабування системи шляхом додавання нових клієнтів або серверів.

Однак, клієнт-серверна архітектура може мати деякі недоліки, зокрема залежність від мережевого з'єднання та можливість перевантаження сервера при великому навантаженні. Правильне проектування та балансування навантаження є важливими аспектами для забезпечення ефективності та доступності системи.

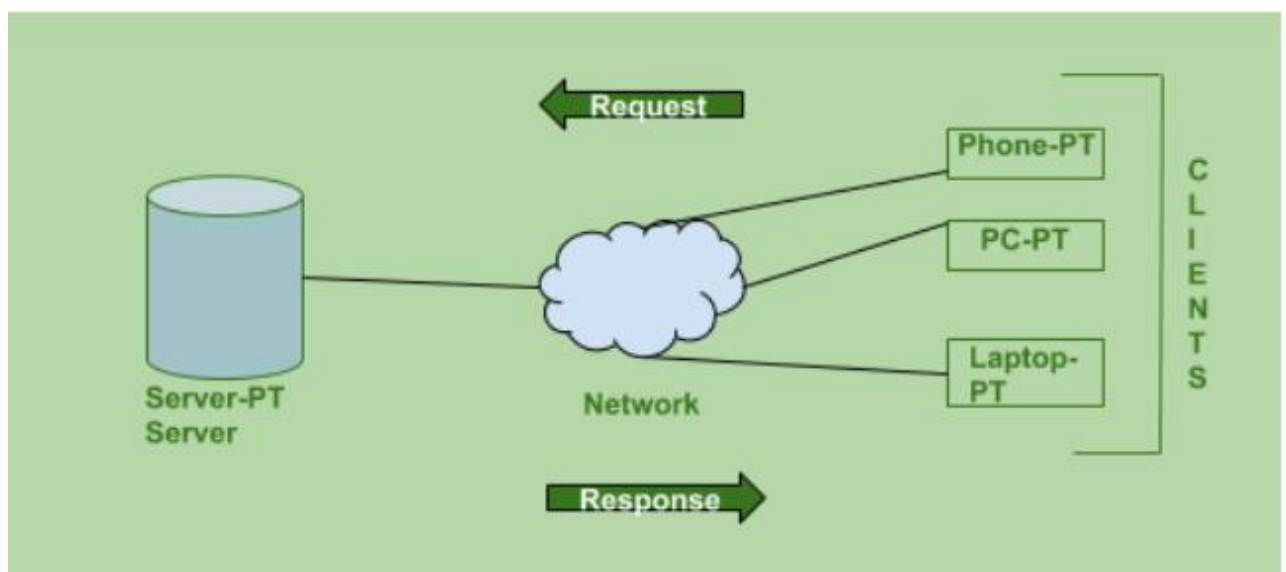


Рисунок 2.2 – Приклад клієнт-серверної архітектури

## 2.4 Вибір технологій та платформи для реалізації додатка-конструктора

Після аналізу принципів проектування додатка-конструктора ми прийшли до висновку, що найбільш оптимальним підходом для нашого проекту є монолітна архітектура. Це обумовлено тим, що у нас є обмежений досвід у створенні додатків, і розробка складних архітектур може бути складною і часоємною задачею. Монолітний підхід дозволяє тримати всю логіку додатку в одному проєкті, без розділення на окремі компоненти або модулі. По аналізу принципів проектування додатка-конструктора, ми вирішили уникнути використання сервер-клієнтної архітектури в нашому випадку. Основна причина полягає у тому, що для серверної частини додатку, яка відповідає б за навчання нейронних мереж та створення датасетів, потрібно було б виділяти додаткові ресурси.

Окрім того, ми обрали мову програмування Python для реалізації нашого додатка-конструктора. Python є простим у сприйнятті і має читабельний синтаксис, що полегшує розробку. Крім того, Python є популярною мовою для розробки нейронних мереж завдяки потужним бібліотекам, таким як TensorFlow і Keras. Ці бібліотеки надають широкі можливості для створення, тренування і застосування нейронних мереж, що відповідає нашим потребам у додатку-конструкторі.

Таким чином, вибір монолітної архітектури та мови програмування Python дозволить нам ефективно розробляти додаток-конструктор, забезпечуючи його простоту у розумінні, структурі та функціональності. Бібліотеки для роботи з нейронними мережами в Python також допоможуть нам досягти поставлених цілей у створенні та навчанні нейронних мереж.

### Висновки по розділу

У даному розділі проведено аналіз методів розробки сучасних десктоп-застосунків з метою вибору оптимальних технологій та проектування архітектури

					ІК93.018БАК.005 ПЗ	Арк.
						26
Зм.	Лист	№ докум.	Підпис	Дата		

для створення власного застосунку. Цей аналіз має велике значення для забезпечення правильного проєктування, оскільки архітектура та використані технології впливають на можливості розширення та підтримки функціоналу майбутнього додатку. Вибір, зроблений на основі розгляду популярних технологій та їхніх можливостей, відображає відповідність застосунку поставленим завданням і потребам користувачів.

У процесі розгляду архітектурних підходів та вибору відповідної архітектури для нашого додатку, було враховано тип, розмір та основний функціонал цього застосунку. Вибір певної архітектури надає структуру проєкту та дозволяє його логічно поділити на окремі компоненти. У нашому випадку було вирішено використовувати монолітну архітектуру. Основними факторами, що вплинули на це рішення, є розмір нашого застосунку на даному етапі розвитку і можливість уникнути додаткових витрат на підтримку, наприклад, окремого сервера для навчання НМ.

Спроектowana система дозволить створювати та розширювати датасети для навчання нейронних мереж. Вона автоматизуватиме процеси створення та розширення датасетів до необхідної кількості, які можуть використовуватися для навчання та валідації моделей. Більш того, систему можна розширити, щоб створювати не лише графічні датасети, а й текстові, звукові та відео, шляхом інтеграції функціоналу для налаштування нейронних мереж під ці типи даних. Це дозволить значно розширити функціонал системи, використовуючи її для завдань, які потребують не лише комп'ютерного зору, а й обробки тексту, розпізнавання мови, генерації текстів, аналізу аудіо та розпізнавання об'єктів на відео, забезпечуючи її корисність у різних сферах застосування.

					ІК93.018БАК.005 ПЗ	Арк.
Зм.	Лист	№ докум.	Підпис	Дата		27

## 3 ОПИС ПРОГРАМНОЇ РЕАЛІЗАЦІЇ

### 3.1 Архітектура програми

Застосунок "Constructor for Dataset Forming and Training NN" є десктоп-рішенням та використовує монолітну архітектуру. Він складається з трьох модулів, які охоплюють всі основні функції формування датасетів та навчання нейронних мереж. Модулі виконують високорівневе розподілення програми на логічні частини, а компоненти поділяють модулі на більш функціональні частини. Важливим принципом розподілу за компонентами є те, що кожен компонент виконує лише одну дію. Це надає унікальну структуру та функціональність додатку.

У розробленому рішенні використовується файлова система як сховище даних. Ми зробили такий вибір, оскільки ФС є простішою у реалізації порівняно з базою даних. Це дозволило нам скоротити час, потрібний для розробки застосунку, і більше уваги приділити іншим компонентам програми. Додатковою перевагою є швидкий доступ до даних. ФС забезпечує швидкий доступ до даних, оскільки дані можуть знаходитись безпосередньо на диску, що дозволяє уникнути додаткової обробки, яка зазвичай відбувається в БД, наприклад, побудова складних запитів.

Використання ФС також дозволяє уникнути залежності від сервера БД, що зменшує складність і вартість інфраструктури. Наприклад, не потрібно ставити окремий сервер для БД або вирішувати питання стабільного з'єднання з нею. Крім того, забезпечення безпеки даних від крадіжки третьою стороною також не вимагає багато зусиль.

ФС є гнучким рішенням, оскільки дозволяє використовувати різні формати та структури даних відповідно до потреб конкретного проєкту. Нарешті, простота створення резервних копій та відновлення є ще однією перевагою використання ФС. Збереження даних у ФС може бути простішим процесом, оскільки файли можна просто копіювати або переміщати у разі втрати даних.

					ІК93.018БАК.005 ПЗ	Арк.
Зм.	Лист	№ докум.	Підпис	Дата		28

### 3.1.1 Створення датасету

Модуль створення датасету є першим компонентом нашого додатку і відіграє важливу роль у процесі формування даних для подальшого використання у навчанні нейронних мереж. Цей модуль включає в себе повний набір операцій, необхідних для ефективного збору, обробки та розширення даних з метою створення якісних датасетів.

Перш за все, модуль створення датасету забезпечує процес збору даних. Це може включати отримання даних з різних джерел, таких як бази даних, веб-скрапінг, API тощо. В нашому випадку використовується API такого рішення, як OIDv4 ToolKit. Збір даних може бути автоматизованим і налаштовуватись залежно від потреб користувача.

Після збору даних модуль проводить обробку цих даних. Це може включати очищення, фільтрацію, нормалізацію та перетворення даних в певний формат або структуру, яка відповідає вимогам навчання нейронних мереж. Обробка даних допомагає покращити якість та узгодженість датасету перед його використанням у навчанні.

Крім того, модуль створення датасету включає функціонал розширення даних. Це означає, що модуль може надавати змогу збільшувати обсяг наявних даних шляхом використання різних модифікаторів. Модифікатори дозволяють виконати такі дії:

- додавати кут повороту;
- віддзеркалення по вертикалі та горизонталі;
- додавати шум;

Це дозволяє покращити різноманітність та репрезентативність датасету, що може позитивно вплинути на якість навчання нейронних мереж.

Усі ці операції збору, обробки та розширення даних виконуються в рамках модуля створення датасету та загалом робота цього модуля виглядає так рисунок 3.1, що дозволяє користувачеві ефективно та зручно формувати датасети, які відповідають конкретним потребам і задачам навчання нейронних мереж. Цей

					ІК93.018БАК.005 ПЗ	Арк.
Зм.	Лист	№ докум.	Підпис	Дата		29

модуль є ключовим етапом у процесі підготовки даних для успішного навчання моделей машинного навчання та глибокого навчання.

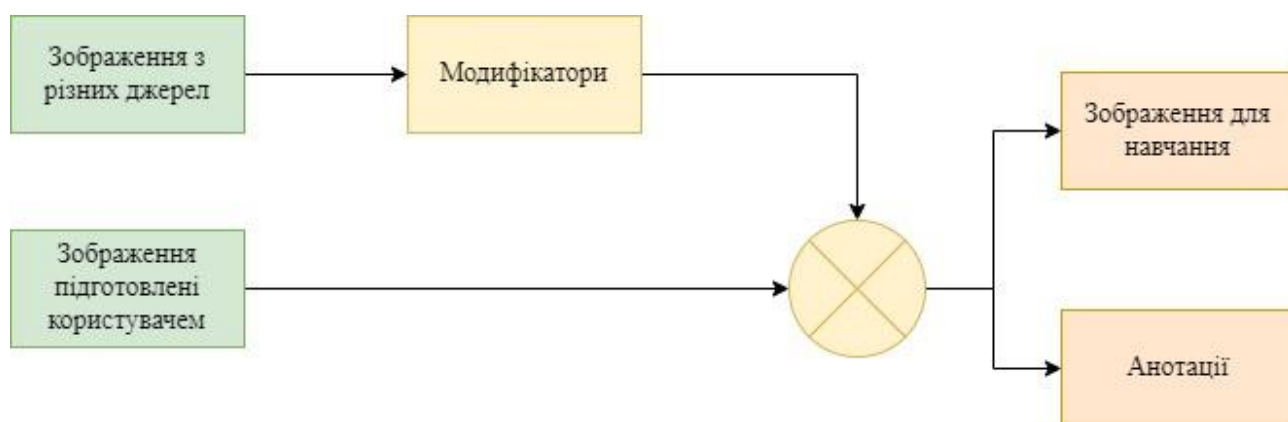


Рисунок 3.1 – Алгоритм роботи модуля створення датасету.

### 3.1.2 Конструктор моделей

Модуль конструктора моделей є іншою складовою частиною нашого додатку, який надає користувачеві можливість створити власну модель нейронної мережі, що відповідає його потребам та вимогам.

Цей модуль забезпечує гнучкий і інтуїтивно зрозумілий інтерфейс, який дозволяє користувачу конструювати модель за допомогою вибору різноманітних шарів та параметрів. Користувач може вибирати і налаштовувати типи шарів, такі як згорткові, повнозв'язані, рекурентні, а також визначати їх розміри, функції активації та інші параметри.

Завдяки модулю конструктора моделей користувач може експериментувати з різними архітектурами та конфігураціями моделей, дозволяючи досягти оптимального балансу між точністю та швидкодією. Крім того, модуль може включати функціонал збереження створених моделей для повернення до них пізніше.

Надзвичайно важливим аспектом модуля конструктора моделей є візуалізація створеної моделі. Користувач може переглядати структуру моделі у інтуїтивно

зрозумілій формі, що дозволяє краще розуміти та налаштовувати модель. Візуалізація може включати графічне представлення шарів та параметрів моделі.

### 3.1.3 Навчання і валідація моделі

Модуль навчання і валідації моделі є третім компонентом нашого додатку, який відповідає за проведення процесу навчання моделі нейронної мережі та оцінку її результатів.

Цей модуль надає користувачу можливість налаштувати параметри навчання, такі як кількість епох, розмір партії (batch size), функцію втрат (loss function) та інші. Користувач може вибрати відповідні значення цих параметрів залежно від специфіки своєї задачі та вимог.

Модуль забезпечує розподілення даних на тренувальні та валідаційні набори. Він автоматично розподіляє дані з датасету на вказані підмножини і забезпечує їх використання під час навчання та валідації моделі. Крім того, модуль може включати функціонал для обробки даних перед навчанням.

Під час навчання модуль здійснює ітерації по тренувальному набору даних та застосовує зазначені алгоритми оптимізації для оновлення ваг моделі. В процесі навчання модуль може відстежувати показники, такі як точність (accuracy) та функцію втрат, щоб користувач міг відслідковувати прогрес навчання та оцінювати якість моделі.

Після завершення навчання модуль проводить валідацію моделі на валідаційному наборі даних. Він обчислює метрики якості, такі як точність та втрати та виводить результат на два графіки. Таким чином користувач може відстежувати тенденцію навчання.

Окрім цього, модуль може забезпечувати можливість зберігання та відновлення навченої моделі, щоб користувач міг використовувати її в майбутньому без необхідності повторювати процес навчання заново. Також є можливість перевірити ефективність навчання на своїх картинках, завантаживши

					ІК93.018БАК.005 ПЗ	Арк.
Зм.	Лист	№ докум.	Підпис	Дата		31

вже навчену модель і бажану картинку і як результат отримати якість передбачення.

## 3.2 Бібліотеки та фреймворки

### 3.2.1 TensorFlow

У розробці частини, відповідальної за навчання нейронних мереж у нашому десктоп-додатку, ми використали TensorFlow – відкрите програмне забезпечення для машинного навчання та глибинного навчання[9].

TensorFlow є бібліотекою, яка надає готові алгоритми чисельних обчислень, виконаних за допомогою графів потоків даних. Ця архітектура базується на використанні вузлів, які відповідають за математичні операції та точки входу/виходу, і ребер, які представляють масиви даних, що перетікають між вузлами у вигляді тензорів.

Одна з особливостей TensorFlow полягає в тому, що вузли можуть бути прикріплені до обчислювальних пристроїв і виконуватися асинхронно, одночасно обробляючи відповідні їм тензори. Це надає можливість організувати паралельну роботу вузлів у нейронній мережі, подібно до одночасної активації нейронів у мозку. Така архітектура дозволяє ефективно використовувати ресурси обчислювальних пристроїв і прискорює процес навчання нейронних мереж. Вона також забезпечує гнучкість та масштабованість, дозволяючи легко змінювати та розширювати структуру моделей. Окрім графів потоків даних та асинхронного виконання, TensorFlow має й інші переваги, які зробили його інструментом для розробки нашого додатку:

– платформна незалежність: TensorFlow підтримує різні операційні системи, такі як Windows, macOS і Linux, а також різні платформи, включаючи настільні комп'ютери, сервери та мобільні пристрої. Це дає розробникам можливість використовувати TensorFlow на різних пристроях без зайвих зусиль;

					ІК93.018БАК.005 ПЗ	Арк.
Зм.	Лист	№ докум.	Підпис	Дата		32

– широка підтримка: TensorFlow підтримує різні мови програмування, включаючи Python, C++, Java та інші. Це дозволяє вам використовувати мову, з якою ви найбільш знайомі, для розробки вашого додатку;

– швидкодія: TensorFlow використовує оптимізовані алгоритми та можливості апаратної підтримки, такі як GPU, що дозволяє досягти високої продуктивності обчислень. Це особливо важливо при навчанні складних моделей на великих наборах даних.

Більше про концепції нейронних мереж ми дізналися з “Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow” автора Aurélien Géron [10].

### 3.2.2 Keras

Для створення моделі і подальшого навчання її необхідно використовувати певний фреймворк або бібліотеку. В нашому випадку ми використали бібліотеку Keras, яка є інтерфейсом вищого рівня до TensorFlow. Keras дозволяє розробникам швидко та легко створювати нейромережі, використовуючи простий та зрозумілий синтаксис [11].

Одна з переваг використання Keras полягає в тому, що вона містить численні готові реалізації популярних будівельних блоків нейромереж, таких як шари (наприклад, повнозв'язні шари, згорткові шари, рекурентні шари), цільові та передавальні функції, оптимізаційні алгоритми тощо. Це дозволяє розробнику значно спростити процес створення моделі та уникнути деталей низькорівневого програмування.

Крім того, Keras надає безліч інструментів для роботи з різними типами даних, зокрема зображеннями та текстом. Це включає в себе підтримку популярних архітектур нейромереж, вбудовані набори даних для швидкого експериментування, а також методи для обробки та попередньої обробки даних. Все це дозволяє значно спростити процес розробки глибоких нейромереж та зосередитися на суттєвих аспектах задачі.

					ІК93.018БАК.005 ПЗ	Арк.
Зм.	Лист	№ докум.	Підпис	Дата		33

### 3.2.3 OIDv4 ToolKit

OIDv4 ToolKit є програмним рішенням, яке було використано для розробки можливості завантаження класів для датасету з інших джерел. Ця програма, написана на мові Python, доступна у відкритому вигляді на платформі GitHub [12].

OIDv4 ToolKit спеціально призначений для роботи з наборами даних з Open Image Dataset. OIDv4 ToolKit дозволяє зручно та автоматично завантажувати ці набори даних з веб-сайту Open Image Dataset безпосередньо у додаток.

Використання OIDv4 ToolKit спрощує процес завантаження та інтеграції додаткових класів для датасету. Користувачу не потрібно вручну завантажувати кожне зображення, якщо він не має вже готового датасету для завантаження. За допомогою OIDv4 ToolKit можна автоматично завантажувати потрібні класи з Open Image Dataset та інтегрувати їх у власний додаток.

#### 3.2.3.1 Open Image Dataset

Open Images – це набір даних, що складається з приблизно 9 мільйонів зображень, які мають анотації на рівні зображення, обмежувальні рамки об'єктів та візуальні зв'язки [13]. У версії V4 навчального набору міститься 14,6 мільйонів обмежувальних рамок для 600 класів об'єктів на 1,74 мільйонах зображень, що робить його найбільшим набором даних з анотаціями місцезнаходження об'єктів. Більшість рамок були вручну намальовані професійними анотаторами для досягнення високої точності та послідовності.

Таблиця 3.1. Виявлення об'єктів

	Навчальні	Валідаційні	Тестові	Класи
Картинки	1,743,042	41,620	125,436	-
Рамки	14,610,229	204,621	625,282	600

Таблиця 3.2. Класифікація зображень

	Навчальні	Валідаційні	Тестові	Класи
Картинки	9,011,219	41,620	125,436	-
Автоматично створені етикетки	78,977,695	512,093	1,545,835	7,870
Етикетки, перевірені людиною	27,894,289	551,390	1,667,399	19,794

Як можна побачити з попередніх таблиць, ми можемо отримати доступ до зображень з трьох різних груп: навчальної, валідаційної та тестової. OIDv4 ToolKit надає можливість вибрати лише певну групу для пошуку.

### 3.2.4 PIL

Наш додаток призначений для роботи з графічними датасетами, тому ми стали перед завданням обробки зображень в програмному середовищі. Для цього ми скористалися бібліотекою Python Imaging Library [14]. PIL надає нам потужні інструменти для роботи з різноманітними аспектами обробки зображень, включаючи зміну розміру, обрізку, корекцію кольору, застосування фільтрів та багато іншого. Використання PIL спрощує нам кодування, необхідне для виконання різних операцій з зображеннями, і дозволяє нам ефективно працювати з графічними даними у нашому додатку.

Однією з особливостей PIL є підтримка роботи з растровими та векторними зображеннями. Він може працювати з одиночними зображеннями, а також з колекціями зображень, що дозволяє нам ефективно опрацьовувати багато зображень одночасно.

PIL також має можливість маніпулювати пікселями зображень, що допомогло нам використовувати шум на зображення для розширення датасету.

Завдяки цим можливостям PIL стає потужним інструментом для роботи з графічними даними у нашому додатку.

### 3.3 Графічний інтерфейс користувача

У минулому, для створення об'єктів дизайну, таких як кнопки, випадаючі списки і меню, потрібно було розробляти їх з нуля. Проте, зараз з'явилися UI фреймворки, які дозволяють спростити і прискорити процес створення програм. Фреймворки вирішують проблему синхронізації внутрішнього стану програми з користувацьким інтерфейсом. Це дозволяє використовувати готові компоненти в своєму проєкті, налаштовуючи їх за необхідністю. Я використав одну з таких UI бібліотек, PyQt5, у своєму десктоп-додатку. Це рішення було унікальним для нашого проєкту, оскільки воно дозволило швидко розробити графічний інтерфейс за допомогою готових компонентів, що значно зекономило час і зусилля. Для детального вивчення Qt було використано книгу “Mastering Qt5” написана Guillaume Lazar та Robin Penea. У цій книзі автори пропонують глибоке дослідження різних аспектів розробки за допомогою Qt5. Вона охоплює широкий спектр тем, включаючи роботу з графікою, роботу з мультимедіа, мережеві операції, взаємодію з базами даних, створення власних віджетів та багато іншого [15].

#### 3.3.1 PyQt

PyQt є одним з найпопулярніших фреймворків для розробки графічного інтерфейсу в мові програмування Python [16]. Він надає потужні можливості для створення професійних і інтерактивних додатків з графічним інтерфейсом користувача. PyQt базується на Qt, крос-платформенному фреймворку для розробки додатків, що працюють на різних операційних системах, таких як Windows, macOS та Linux. Використовуючи PyQt можна створювати додатки, які

					ІК93.018БАК.005 ПЗ	Арк.
Зм.	Лист	№ докум.	Підпис	Дата		36

працюють на різних платформах без необхідності писати окремий код для кожної з них.

Основна перевага PyQt полягає в його повній інтеграції з мовою Python. Завдяки чому можна використовувати всю потужну функціональність Python разом з можливостями Qt для створення розширених графічних інтерфейсів. PyQt надає доступ до широкого спектру графічних елементів, таких як кнопки, поля введення, таблиці, діаграми та багато іншого, що допомогло нам створити багатофункціональний додаток зі зручним інтерфейсом. Крім того, PyQt має велику спільноту розробників, яка активно підтримує цей фреймворк. Де можна знайти багато прикладів коду, документацію і форуми, які допомогли нам вирішити будь-які питання або проблеми, що виникали під час розробки.

### 3.3.2 Qt Designer

Qt Designer – це графічний інструмент для створення та редагування графічних інтерфейсів користувача (GUI) від компанії Qt. Він надає зручне інтуїтивно зрозуміле середовище для швидкого розроблення складних та красивих вікон. Qt Designer дозволив нам візуально створювати GUI за допомогою перетягування та розміщення елементів у вікні дизайнера (рисунок 3.1). Працюючи можна вибрати з великого набору готових елементів, таких як кнопки, поля введення, таблиці, меню та багато інших, а також налаштовувати їх властивості та взаємодію. Після створення дизайну інтерфейсу у Qt Designer, можна зберегти його і підключити до свого коду Python за допомогою PyQt. Це дозволило нам використовувати створений нами інтерфейс у нашому дипломному проєкті, роблячи його більш інтуїтивно зрозумілим та професійним.

Отже Qt Designer спростив процес створення складних і красивих вікон для нашого дипломного проєкту, дозволивши нам швидко і ефективно втілити наші ідеї в реальність.

					ІК93.018БАК.005 ПЗ	Арк.
Зм.	Лист	№ докум.	Підпис	Дата		37

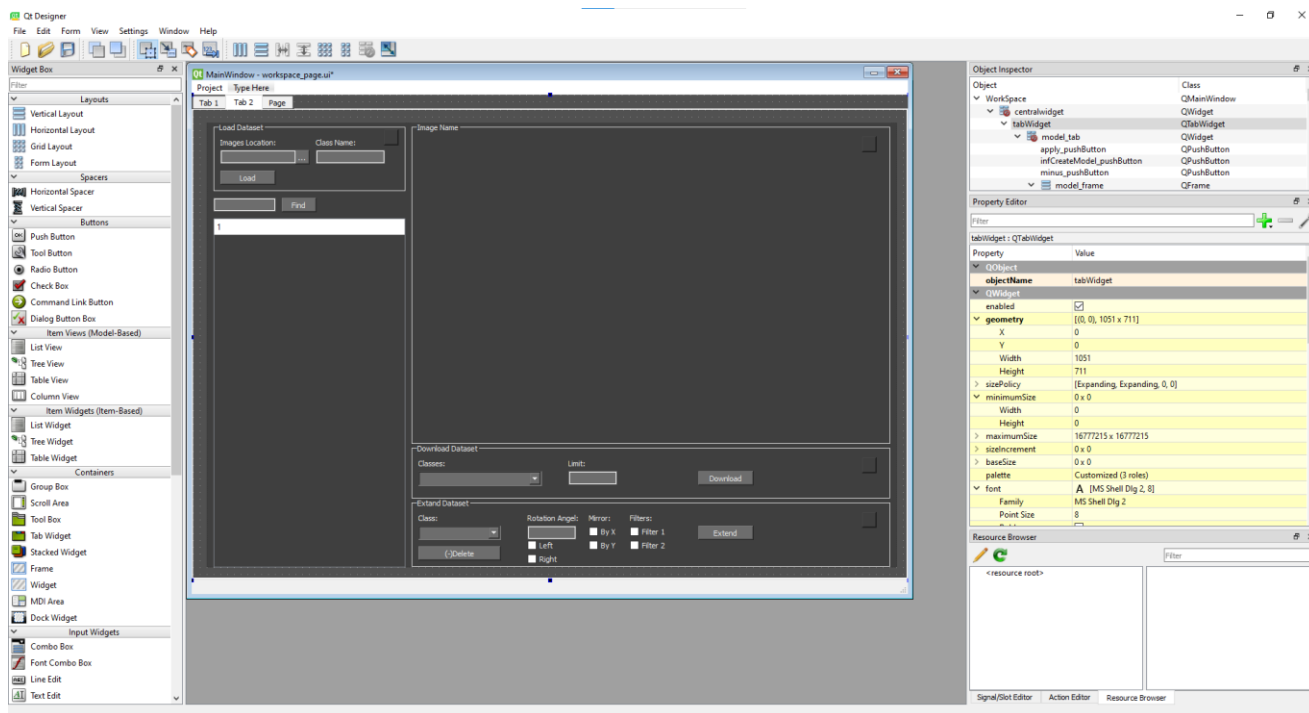


Рисунок 3.2 – Приклад інтерфейсу Qt Designer

### 3.4 Мова програмування

Для вибору мови програмування важливо чітко розуміти функціональні вимоги проєкту. Деякі завдання можуть бути зручно реалізовані в одній мові, тоді як для інших може бути краще використовувати іншу мову. У моєму випадку, обрав мову програмування Python для розробки десктоп-додатку з метою створення датасетів та навчання глибоких нейронних мереж. Я обрав Python через його універсальність і зручність для роботи з нейронними мережами. Python надає широкі можливості для обробки даних, машинного навчання та наукових обчислень, що допомагає зробити розробку ефективною і продуктивною [17].

#### 3.4.1 Python

Python – це інтерпретована мова програмування, що означає, що код виконується рядок за рядком без необхідності компіляції перед запуском. Це має свої переваги в контексті дипломного проєкту, оскільки дозволяє швидше

розробляти та тестувати різні частини програми без затримок, пов'язаних з компіляцією [18].

Python є об'єктно-орієнтованою мовою програмування, що дозволяє організувати код у вигляді об'єктів, які містять дані та функції, що з ними пов'язані. Це сприяє структурованості та повторному використанню коду, особливо при розробці складних програмних проєктів. В рамках дипломного проєкту використання об'єктно-орієнтованого підходу може сприяти створенню модульної та легко розширюваної системи.

Перевагами використання Python є:

– простота вивчення і використання: Python має простий і зрозумілий синтаксис, що робить його легким для вивчення навіть для початківців. Він також має велику кількість документації та активну спільноту розробників, яка готова надати допомогу і підтримку;

– багатофункціональність: Python має велику кількість стандартних бібліотек і модулів, що дозволяють виконувати різноманітні завдання без необхідності великого обсягу коду. Наприклад, в ньому є модулі для роботи з файлами, мережевими операціями, обробки тексту, веб-розробки, наукових обчислень та багато іншого;

– платформонезалежність: Python працює на різних операційних системах, включаючи Windows, macOS і Linux. Це означає, що ваш дипломний проєкт може бути виконаний на будь-якій платформі без необхідності вносити значні зміни в код;

– широке застосування: Python широко використовується у багатьох сферах, включаючи веб-розробку, наукові дослідження, обробку даних, робототехніку, ігрову розробку та багато іншого. Це означає, що навички, отримані при роботі з Python, можуть бути корисними для подальшої кар'єри.

Недоліки Python:

– швидкодія: Python є інтерпретованою мовою, що може призвести до меншої швидкодії порівняно з компільованими мовами, такими як C++. У

					ІК93.018БАК.005 ПЗ	Арк.
Зм.	Лист	№ докум.	Підпис	Дата		39

випадку обробки великих обсягів даних або вимоги до високої продуктивності, Python може бути менш оптимальним вибором;

– обмежені можливості розробки віконного додатку: Python не є найкращим вибором для розробки десктопних додатків. Хоча є інструменти, такі як PyQt і Tkinter, які дозволяють розробляти десктопні додатки на Python, їх можливості можуть бути обмежені порівняно з іншими мовами програмування;

– пам'ять та ресурси: Python може вимагати більше пам'яті та ресурсів у порівнянні з іншими мовами програмування. Це особливо помітно при роботі з великими обсягами даних або при використанні певних бібліотек, які можуть бути важкими для системи.

### 3.4.2 PyCharm

Для нашого проекту ми обрали PyCharm – потужне інтегроване середовище розробки (IDE), спеціально створене для програмування на мові Python [19]. Воно надає набір розширених інструментів і функцій, які значно спрощують процес розробки, налагодження та тестування коду. За допомогою PyCharm ми змогли створити та керувати віртуальними середовищами, що дозволило нам ізолювати проект і його залежності, уникнути конфліктів між пакетами та забезпечити чистоту нашого розробкового середовища.

PyCharm також забезпечує потужні можливості автодоповнення коду, що допомогло нам писати код швидше та уникати помилок. Воно виявляє потенційні проблеми в коді, надає підказки та попередження, що покращує якість і надійність нашого програмного забезпечення. Крім того, PyCharm має вбудовану підтримку системи контролю версій Git, що дозволило нам зручно виконувати коміти, відстежувати зміни.

Завдяки розширеному набору інструментів для налагодження, PyCharm надало нам зручність при виявленні та виправленні помилок у процесі розробки. Можливість ставити точки зупинки, виконувати крок за кроком та спостерігати змінні допомагає нам ефективно налагоджувати код і забезпечувати його

					ІК93.018БАК.005 ПЗ	Арк.
Зм.	Лист	№ докум.	Підпис	Дата		40

безперебійну роботу. В цілому, використання PyCharm значно підвищило продуктивність та якість нашого розробку, забезпечило комфортне та ефективне середовище для роботи з кодом і сприяло успішній реалізації нашого дипломного проєкту.

### 3.5 Програмні рішення та розроблені алгоритми

#### 3.5.1 Початкова сторінка

Для зручності розділення робочої частини програмного коду від файлів додатку, було прийнято рішення створювати проєкти для користувача. Проєкт – це набір папок, в яких зберігаються датасети та моделі нейронних мереж. Для реалізації цього механізму було розроблено початкове вікно, де користувач може створити новий проєкт, відкрити вже існуючий проєкт та мати можливість швидкого доступу до певного проєкту за допомогою ярликів. Це дозволяє ефективно управляти різними проєктами та зберігати їх окремо, спрощуючи навігацію та організацію робочого процесу.

Ми взяли за приклад початкове вікно в середовищі розробки PyCharm (рисунок 3.3). У цьому вікні ми бачимо функцію "Створити новий проєкт", при натисканні на яку відкривається нове вікно (рисунок 3.4), де користувач може ввести шлях, де буде зберігатися новий проєкт, його назву, а також використовувати багато інших функцій, що надає PyCharm. Наступна функція, яку ми можемо спостерігати, це кнопка "Відкрити", яка відповідає за відкриття існуючих проєктів. Для цього використовується діалогове вікно (рисунок 3.5), де користувач повинен знайти проєкт на своєму комп'ютері. І остання функція, яку ми взяли за приклад (але не остання загалом в PyCharm), це "Останній проєкт", який дозволяє користувачеві легко відкрити проєкт простим кліком на ярлик.

Ми також розробили власний алгоритм, який забезпечує безпеку та коректність роботи нашого додатку. Цей алгоритм використовується при створенні нового проєкту, де ми формуємо файл з закодованою сигнатурою. При відкритті проєкту з збереженого носія додаток перевіряє наявність цього файлу та

					ІК93.018БАК.005 ПЗ	Арк.
Зм.	Лист	№ докум.	Підпис	Дата		41

порівнює його сигнатуру, розкодовуючи її. На основі результату цієї перевірки програма робить висновки, чи слід відкривати цей проєкт. Важливо зазначити, що цей алгоритм не є ідеальним і можливі спроби обходу його захисту. З цим уявленням ми маємо плани на майбутнє для розширення цього алгоритму та покращення його безпеки та ефективності. Приклад роботи алгоритму зображений на блок схемі (рисунок 3.6):

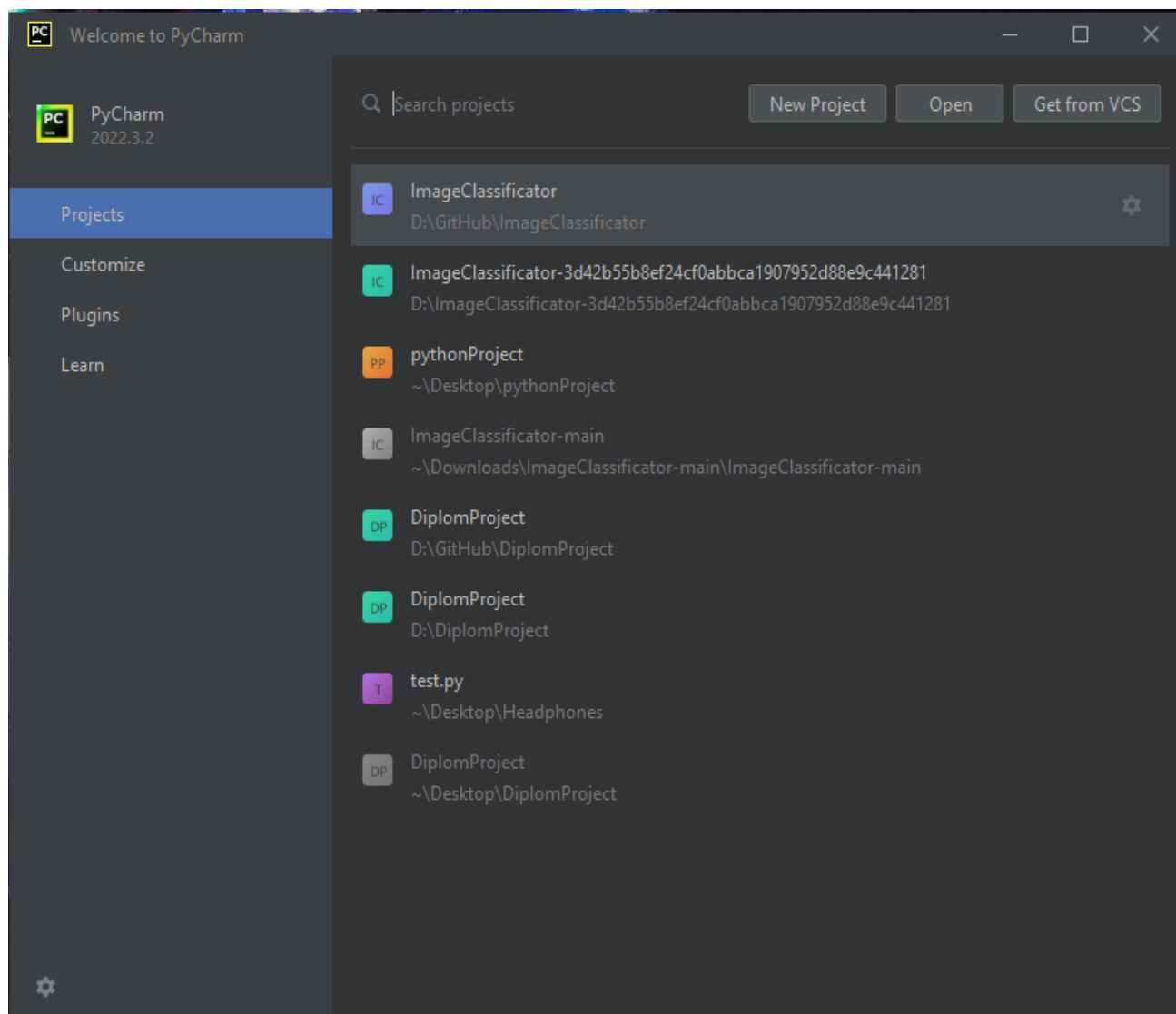


Рисунок 3.3 – Початкове вікно PyCharm

					ІК93.018БАК.005 ПЗ	Арк.
Зм.	Лист	№ докум.	Підпис	Дата		42

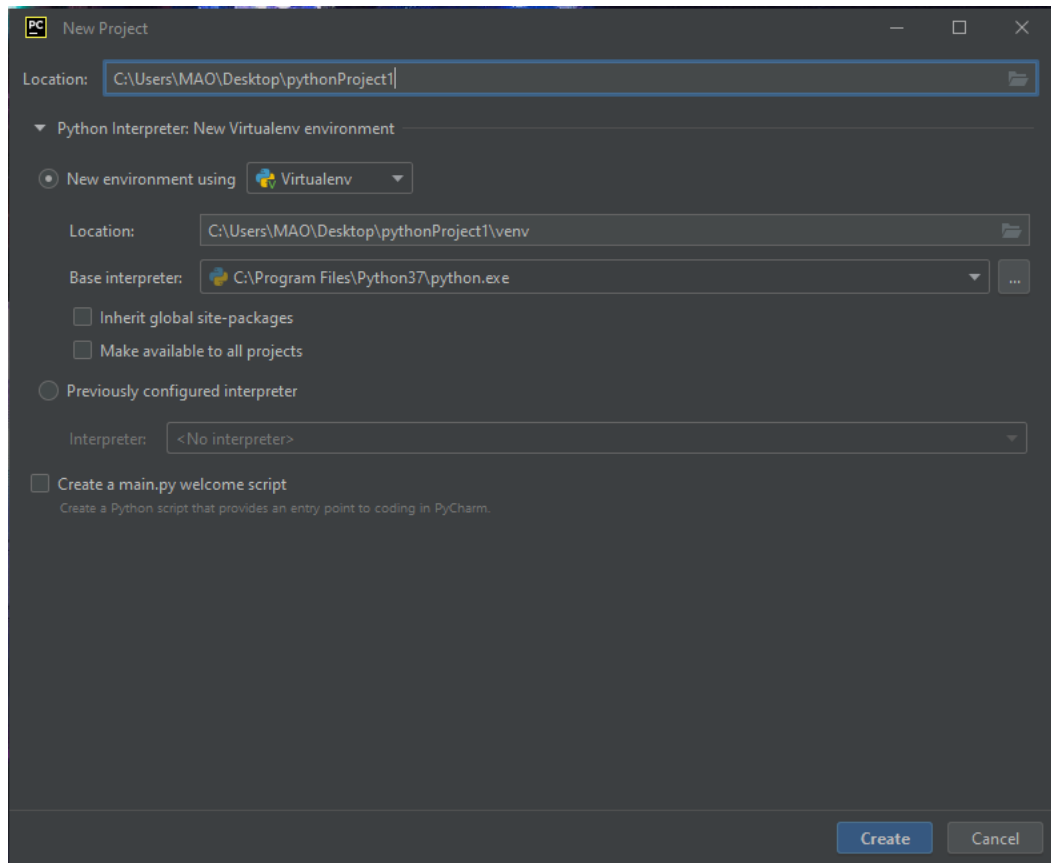


Рисунок 3.4 – Вікно "Створити новий проект"

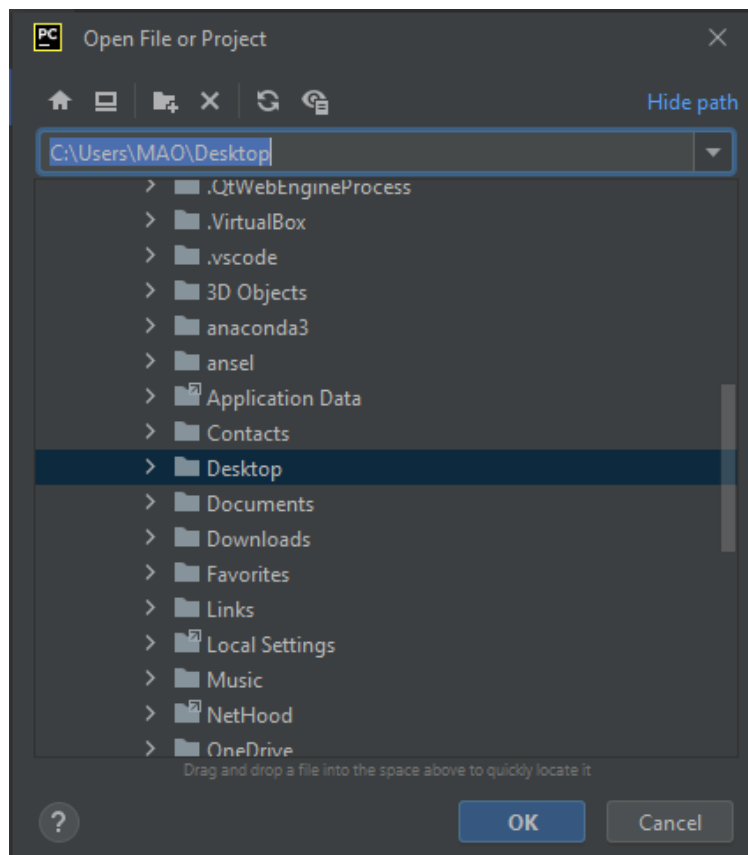


Рисунок 3.5 – Діалогове вікно "Відкрити"

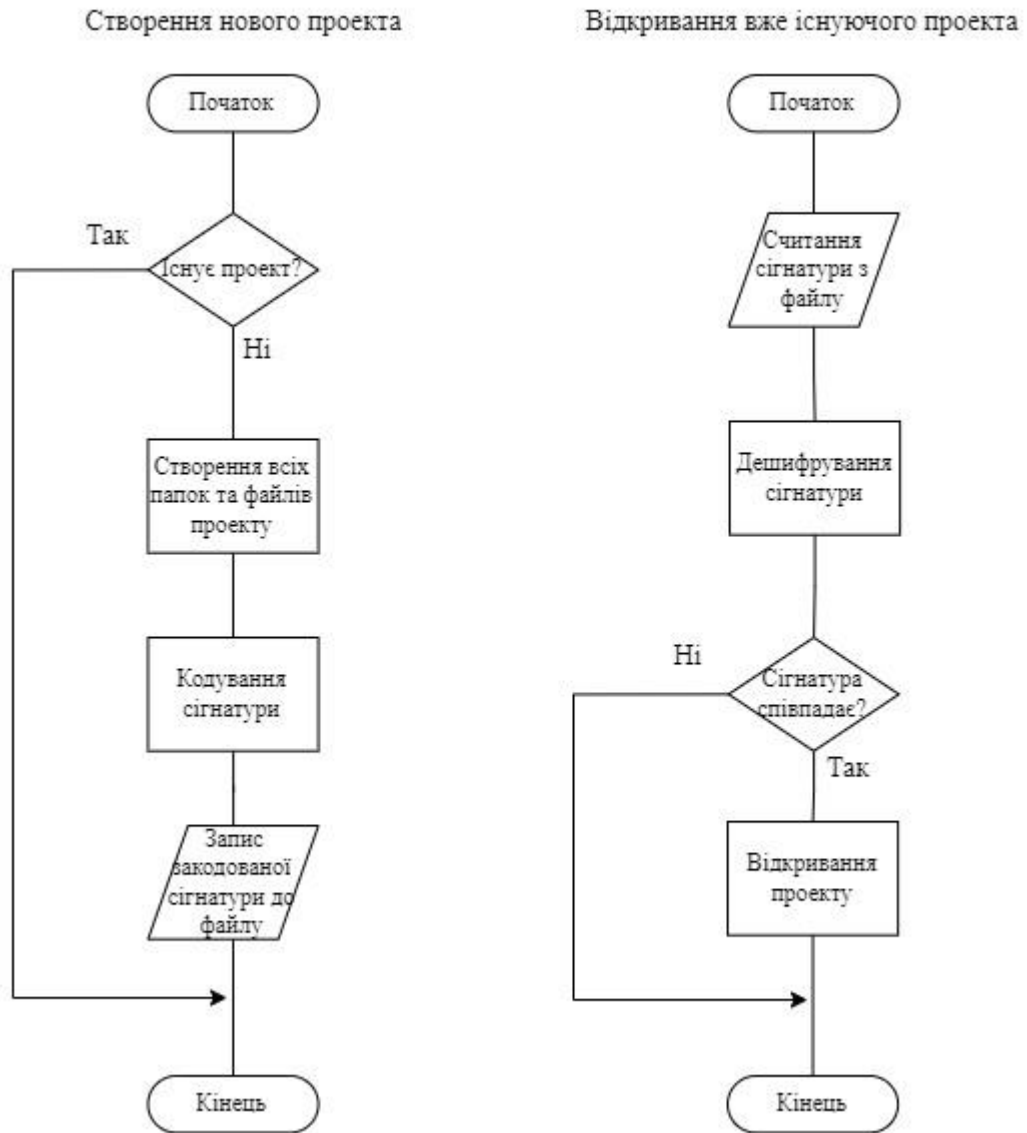


Рисунок 3.6 – Блок схема алгоритму створення та відкривання проекту

### 3.5.2 Структура проекту користувача

Структура нашого проекту включає три основні папки, чотири підпапки та один файл зі сигнатурою.

Основні папки:

– папка "Датасети": В цій папці зберігаються всі датасети, які використовуються в проекті. Вона дозволяє організувати та зберігати дані, необхідні для тренування та тестування нейронних мереж;

Зм.	Лист	№ докум.	Підпис	Дата

– папка "Моделі": Тут знаходяться всі моделі нейронних мереж, які використовуються в проєкті. Ця папка містить код, параметри та ваги моделей, що дозволяє їх зберігати та використовувати для прогнозування;

– папка "Завантаження": В цій папці знаходяться оригінальні файли, які були завантажені з Open Image Dataset. Це початковий набір даних, з якими ми працюємо.

Підпапки:

– папка "Тренувальні": Ця папка містить оброблені файли зі зменшеними розмірами та зміненим форматом, які будуть використовуватися для навчання. Розміри зображень скорочуються до 256x256, а кількість каналів змінюється на 3;

– папка "Тренувальні + фільтри": В цій папці знаходяться остаточні датасети, які були розширені до необхідної кількості, використовуючи фільтри. Ці файли готові для використання в процесі навчання моделей;

– папка "Пресети": У цій папці зберігаються пресети моделей, які можуть бути завантажені в конструкторі моделей. Вони включають попередньо налаштовані параметри та налаштування моделей для зручної розробки;

– папка "Навчені": В цій папці знаходяться навчені моделі та збережений код моделей. Вона включає файли, які представляють навчені ваги моделей та код, який був використаний для їх тренування.

Ця структура дозволяє зручно організувати файли та дані в проєкті, забезпечуючи логічну структуру та легку навігацію між різними компонентами. Графічне відображення структури проєкту зображено на рисунку 3.7:

					ІК93.018БАК.005 ПЗ	Арк.
Зм.	Лист	№ докум.	Підпис	Дата		45

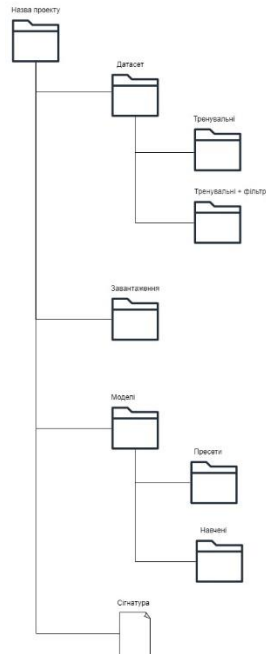


Рисунок 3.7 – Структура проєкта

### 3.5.3 Алгоритм збереження моделі

Під час розробки нашого додатку виникали ситуації, коли нам потрібно було знайти рішення для реалізації певної функціональності. Одним з таких випадків була потреба в створенні механізму для збереження моделі окремо від процесу навчання НМ. Нам потрібно було знайти спосіб, як створити модель, що буде використовуватися під час навчання.

Для вирішення цього питання було прийняте рішення розробити генератор скрипта. Цей генератор буде створювати модель, яку ми зможемо використовувати у нашому проєкті. З його допомогою ми зможемо генерувати код для створення та налаштування моделі відповідно до наших потреб. Таким чином, ми отримаємо окремий скрипт, який буде містити налаштування моделі та може бути використаний в навчанні НМ.

Алгоритм цього рішення базується на використанні звичайного перебору if-else, що аналогічний конструкції switch case в інших мовах програмування. Алгоритм полягає у тому, що я проходжуся по кожному шару по порядку та перевіряю його тип. Для кожного типу шару я написав відповідний рядок коду в скрипті. За допомогою if-else конструкцій я перевіряю тип кожного шару і

					ІК93.018БАК.005 ПЗ	Арк.
Зм.	Лист	№ докум.	Підпис	Дата		46

виконую відповідну дію, яка потрібна для створення та налаштування моделі. Таким чином, я генерую скрипт, який містить налаштування для кожного типу шару. Приклад роботи алгоритму зображений на блок схемі рисунок 3.8:

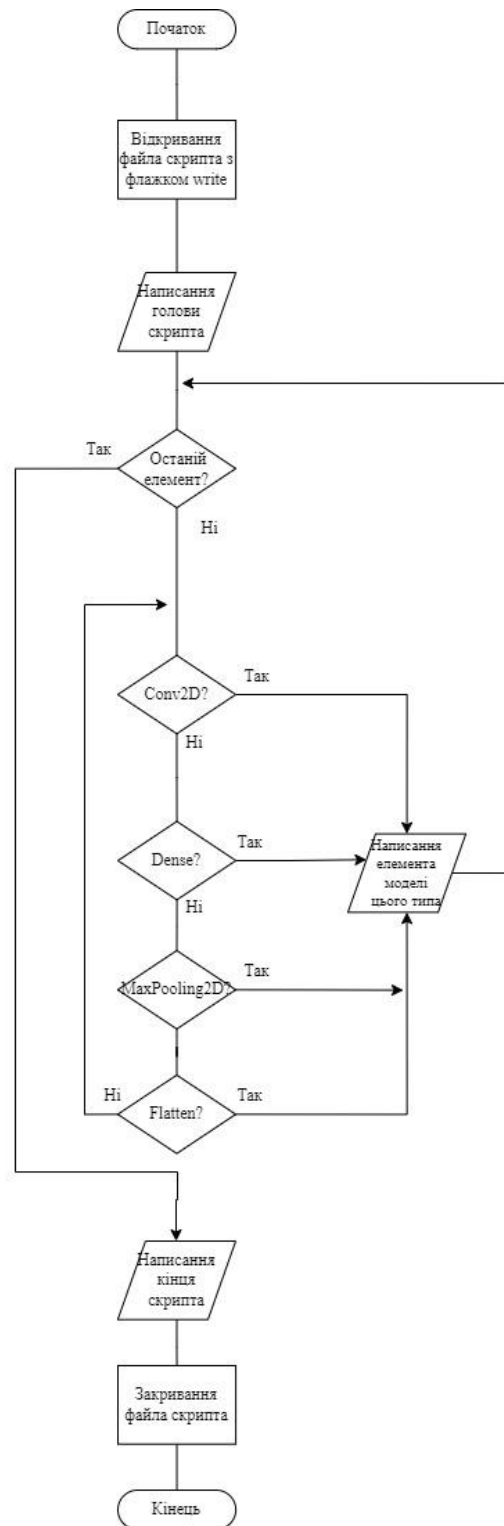


Рисунок 3.8 – Блок схема алгоритму генерації скрипту моделі

### 3.5.4 Алгоритми розширення датасету

Для можливості розширення було розроблено алгоритм, який базується на умовних операторах (if-else) для перевірки флажків кута повороту, віддзеркалення та застосування шуму.

Для реалізації кута повороту та віддзеркалення використовувався функціонал бібліотеки PIL. У випадку віддзеркалення достатньо було просто повернути зображення без змін, але у випадку повороту на кут, який не є кратним 90 градусам, виникали чорні зони, які могли негативно вплинути на навчання. Для уникнення цього було розроблено рішення, яке обрізало зображення, зберігаючи найбільшу площу без повних чорних елементів.

Для застосування шуму був розроблений алгоритм, що включає такі кроки:

- застосування покращення контрастності до зображення;
- проходження по кожному пікселю зображення;
- отримання значень RGB каналів пікселя;
- генерація випадкового значення в межах від -50 до 50;
- додавання згенерованого шуму до значень каналів пікселя та оновлення пікселя зображення.

Ці дії дозволяють застосовувати шум до зображень і отримувати розширені варіанти для подальшого використання.

### Висновки по розділу

У третьому розділі ми розробили монолітну архітектуру десктоп-додатка, яка включає три основних модуля: створення датасету, конструктор моделей і навчання й валідацію нейронної мережі. Для навчання нейронної мережі були використані бібліотеки TensorFlow і Keras, а для роботи з зображеннями – PIL. Також було інтегровано програмне рішення OIDv4 ToolKit, яке забезпечило збір картинок з джерела Open Image Dataset.

					ІК93.018БАК.005 ПЗ	Арк.
Зм.	Лист	№ докум.	Підпис	Дата		48

При розробці графічного інтерфейсу користувача (GUI) використовувався середовище QtDesigner, де були створені прототипи інтерфейсів додатка. Після цього вони були вбудовані та редаговані за допомогою бібліотеки PyQt.

Мовою програмування, яку було обрано як основну, є Python, оскільки більшість додатків, пов'язаних з нейромережами, розроблені на цій мові. Також для розробки використовувалось середовище PyCharm.

					ІК93.018БАК.005 ПЗ	Арк.
Зм.	Лист	№ докум.	Підпис	Дата		49

## 4 РОБОТА КОРИСТУВАЧА З ДОДАТКОМ

### 4.1 Інструкція користувача

Для початку роботи з десктоп-додатком "Constructor for Dataset Forming and Training NN" потрібно створити проєкт на початковому екрані (рисунок 4.1).

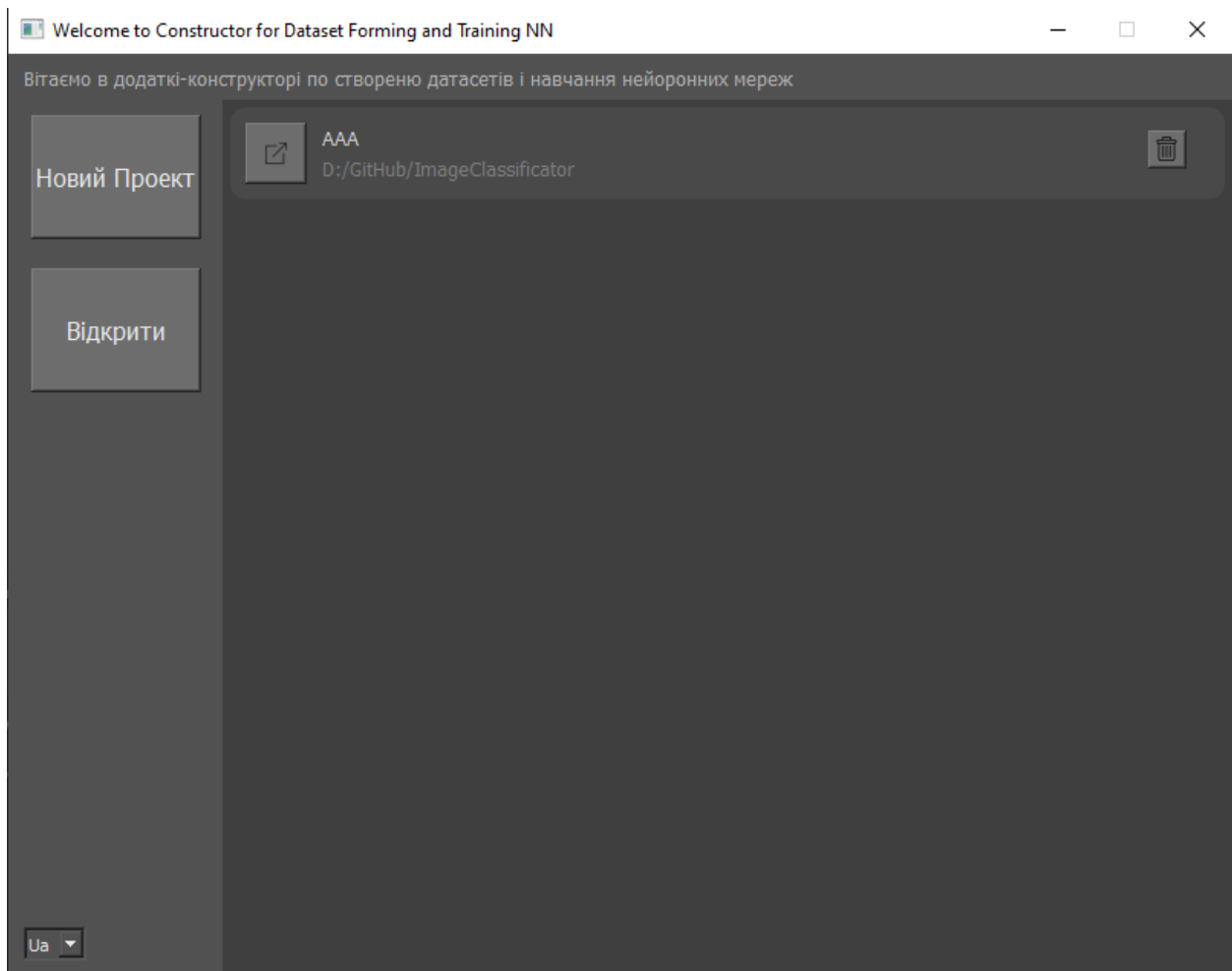


Рисунок 4.1 – Початкове вікно додатку

Після входу у додаток є 3 варіанти продовження роботи:

– якщо користувач вже створив проєкт в додатку, він може натиснути на ярлик й увійти в поточний проєкт;

– якщо користувач ще не створив проєкт, він може натиснути кнопку "Новий Проєкт". Після цього відкриється вікно, де користувач може вибрати місце розташування для майбутнього проєкту та ввести назву проєкту;

					ІК93.018БАК.005 ПЗ	Арк.
Зм.	Лист	№ докум.	Підпис	Дата		50

– якщо необхідно відкрити сторонній проєкт, користувач натискає на кнопку “Відкрити” та знаходить його в діалоговому вікні;

–також можливо змінити мову додатку

Розпочнемо з вікна створення нового проєкту. Відкривши вікно створення нового проєкту, користувач зможе вказати розташування для майбутнього проєкту і ввести його назву (рисунок 4.2). Після успішного створення проєкту він буде автоматично відкритий, в іншому випадку ви отримаєте повідомлення про помилку.

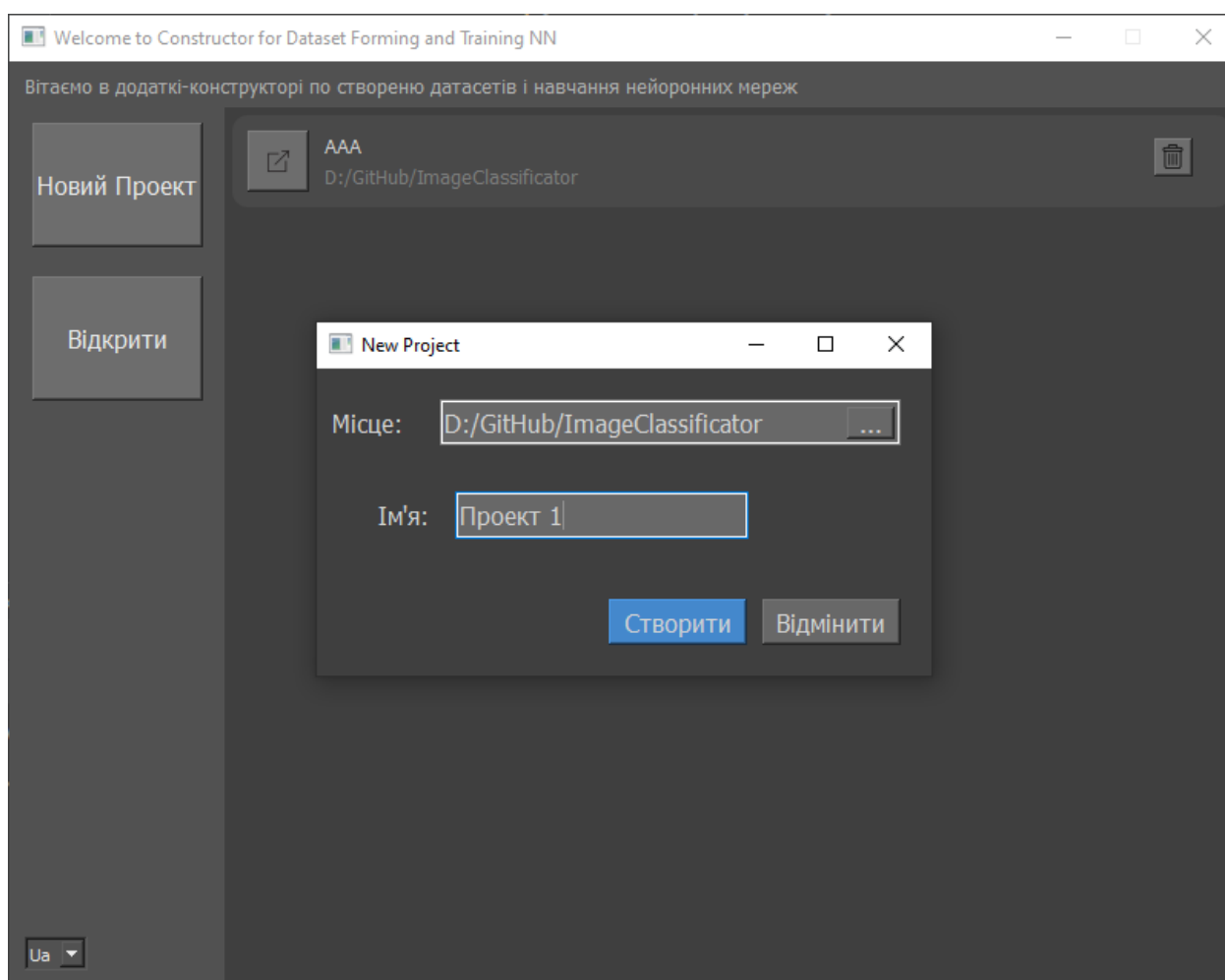


Рисунок 4.2 – Вікно створення нового проєкту

У випадку, якщо користувач вже створив проєкт – він може натиснути на ярлик і відкрити цей проєкт (рисунок 4.3).

					ІК93.018БАК.005 ПЗ	Арк.
Зм.	Лист	№ докум.	Підпис	Дата		51



Рисунок 4.3 – Ярлик нещодавніх проєктів

Якщо користувач бажає відкрити проєкт, який він не створював, він може натиснути кнопку "Відкрити" і обрати його у діалоговому вікні (рисунок 4.4). Якщо проєкт належить нашому додатку, він буде відкритий негайно. В іншому випадку з'явиться повідомлення про помилку, що цей проєкт не належить даному додатку.

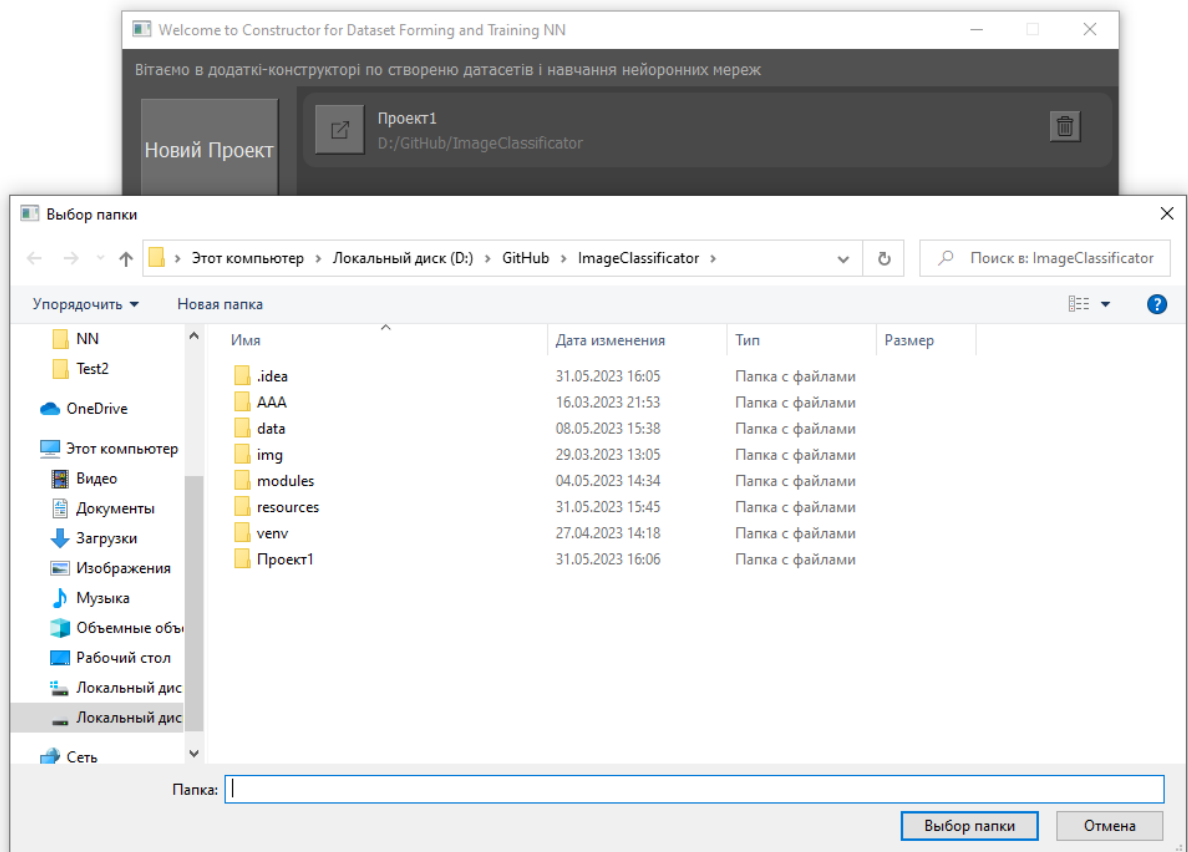


Рисунок 4.4 – Діалогове вікно вибору проєктів

Потрапивши на робочий стіл додатку (рисунок 4.5), користувач може створити модель в конструкторі, працювати з датасетом та навчити або перевірити НМ. Використовуючи вкладки користувач може перейти до:

- конструктору моделей;
- вікна створення датасетів;
- вікна навчання та валідації НМ.

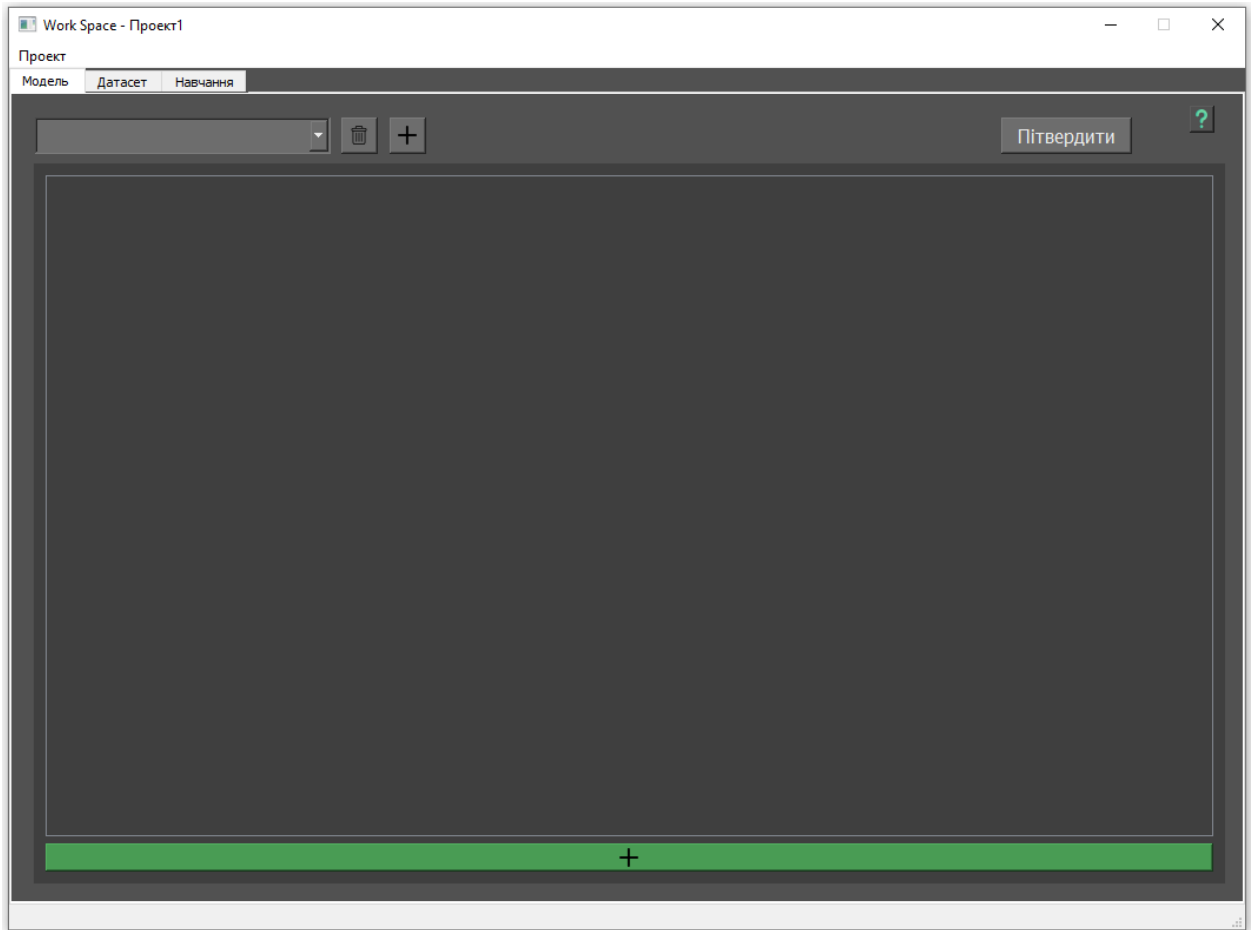


Рисунок 4.5 – Вікно робочого стола

Почнемо роботу з підготовки датасету для навчання. Перейшовши до вкладки "Датасет" (рисунок 4.6), користувач може побачити наступні функції:

- завантаження датасету: Тут користувач може завантажити картинки зі свого носія та призначити їм клас;
- дерева файлових систем, що відображають папки проєкту "train" та "train+filter";
- велике поле, де можна працювати з кожною картинкою окремо;
- скачати Датасет: Тут користувач може завантажити потрібні класи та кількість картинок з Open Image Dataset;

					ІК93.018БАК.005 ПЗ	Арк.
Зм.	Лист	№ докум.	Підпис	Дата		53

– розширення Датасету: Тут користувач може розширити певний клас до потрібної кількості шляхом додавання куту повороту, віддзеркалення та застосування шуму.

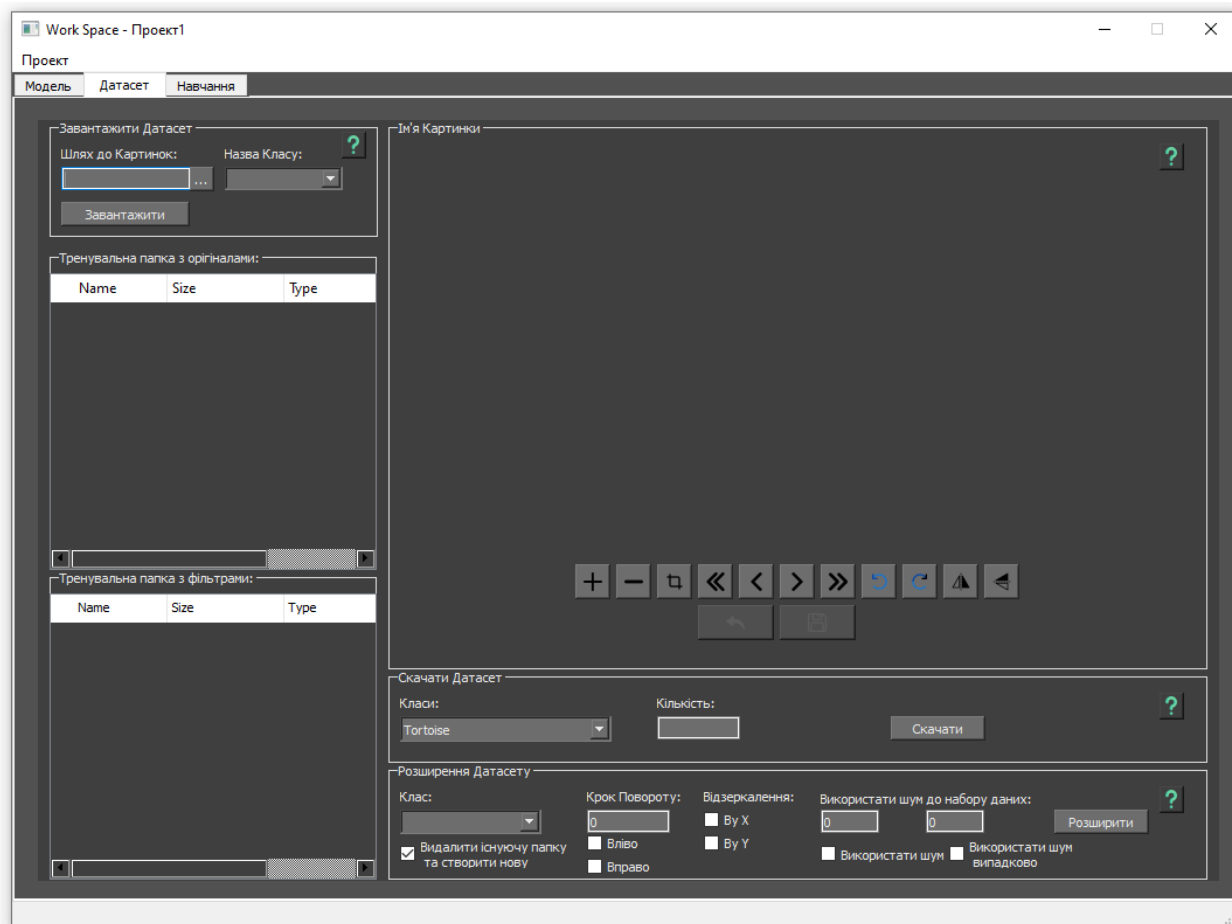


Рисунок 4.6 – Вікно роботи з датасетом

Почнемо з першого кроку. Якщо користувач вже має фотографії на своєму носії, наприклад, рентгенівські знімки клітин, де одна клітина уражена раковою пухлиною, а інша – ні, він може натиснути кнопку "..." і вибрати папку через діалогове вікно, як показано на рисунку 4.4, з папкою, що містить уражені клітини. Після цього він може присвоїти цій папці клас "Уражена" і натиснути "Завантажити". Ці зображення автоматично завантажуться до папок "train" і "train+filter".

Існує інший спосіб завантаження класів з Open Image Dataset. Для цього потрібно вибрати бажаний клас зі списку класів у випадяючому меню "Класи".

Потім в поле "Кількість" введіть бажану кількість зображень, яку потрібно завантажити. Після цього натисніть кнопку "Скачати", і додаток запустить OIDv4 на фоні (рисунку 4.7), де ви зможете відстежувати прогрес завантаження. Після успішного завантаження ці зображення автоматично з'являться у папках "train" і "train+filter".

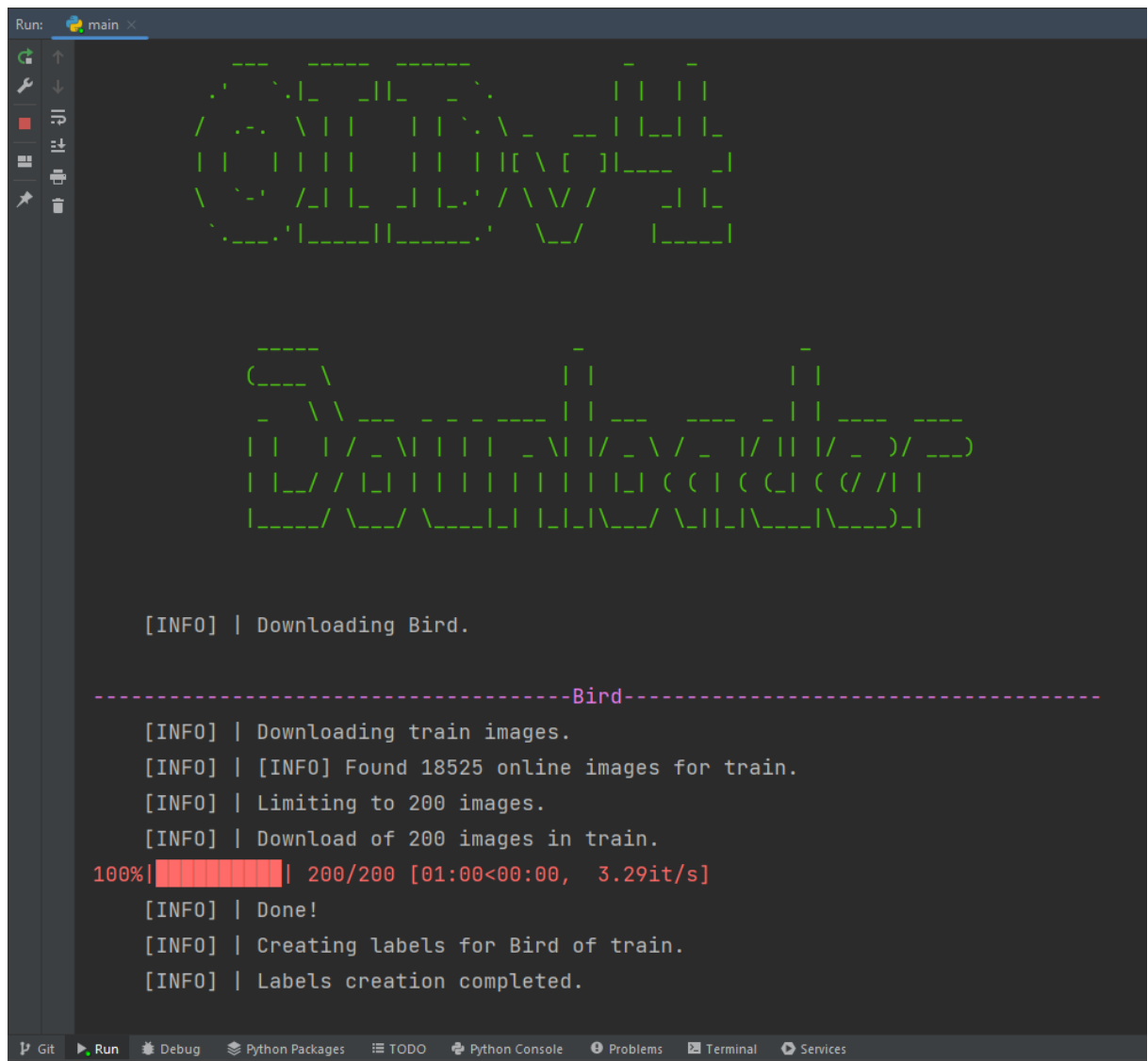


Рисунок 4.7 – Приклад роботи програми OIDv4 ToolKit

Для відстежування класів і їх наповнення використовується дерево файлової системи (рисунку 4.8). За допомогою цих дерев користувач може переглядати класи, їх склад та виконувати операцію по видаленню класів або окремих зображень.

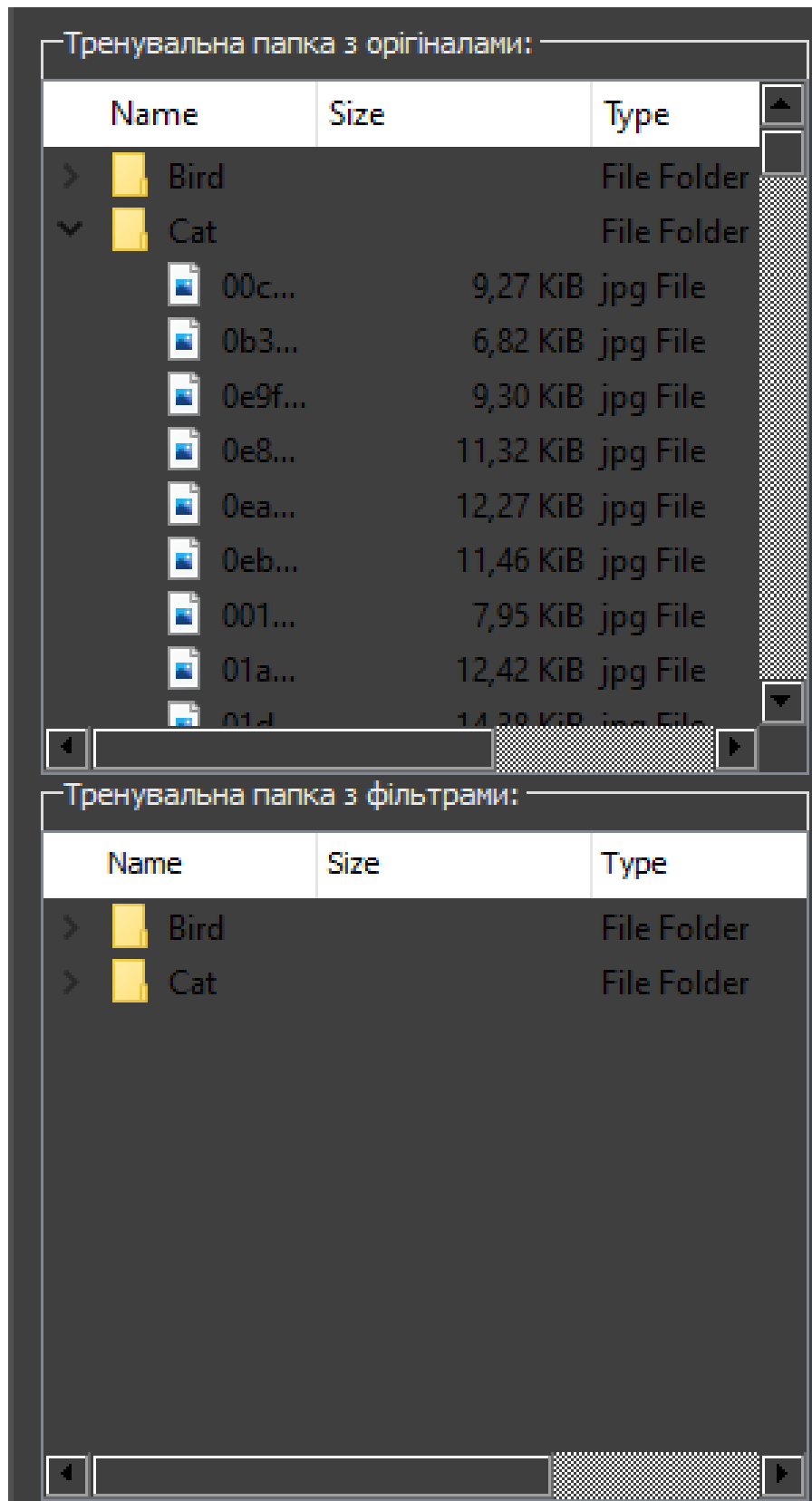


Рисунок 4.8 – Дерева файлових систем

Кожну картинку можна також редагувати окремо. Для цього необхідно клацнути на клас у дереві або безпосередньо на картинку. Після цього зображення

відкриється у великому блоку, подібно до рисунка 4.9, де доступні 14 кнопок – 12 основних та 2 допоміжні. Дві пари кнопок відповідають за трансформацію зображень, зокрема поворот вправо/вліво на 90 градусів та відзеркалення по горизонталі та вертикалі. Також є кнопки для переміщення між зображеннями у межах класу – вперед та назад, а також кнопки для переходу на початок та в кінець класу. Останні три головні кнопки дозволяють додавати одне зображення до поточного класу за допомогою діалогового вікна, видаляти вибране зображення з класу та виконувати обрізку. Функція обрізки активується натисканням третьої кнопки зліва, після чого режим виділення стає доступним і дві допоміжні кнопки стають активними. Користувач може виділити область за допомогою перетягування мишкою, подібно до рисунка 4.10, яку потрібно обрізати. Після обрізання є два варіанти дії: збереження за допомогою нижньої правої кнопки, після натискання якої допоміжні кнопки стануть неактивними, результат буде збережено, а бо ліва нижня, яка повертає до початкового стану.

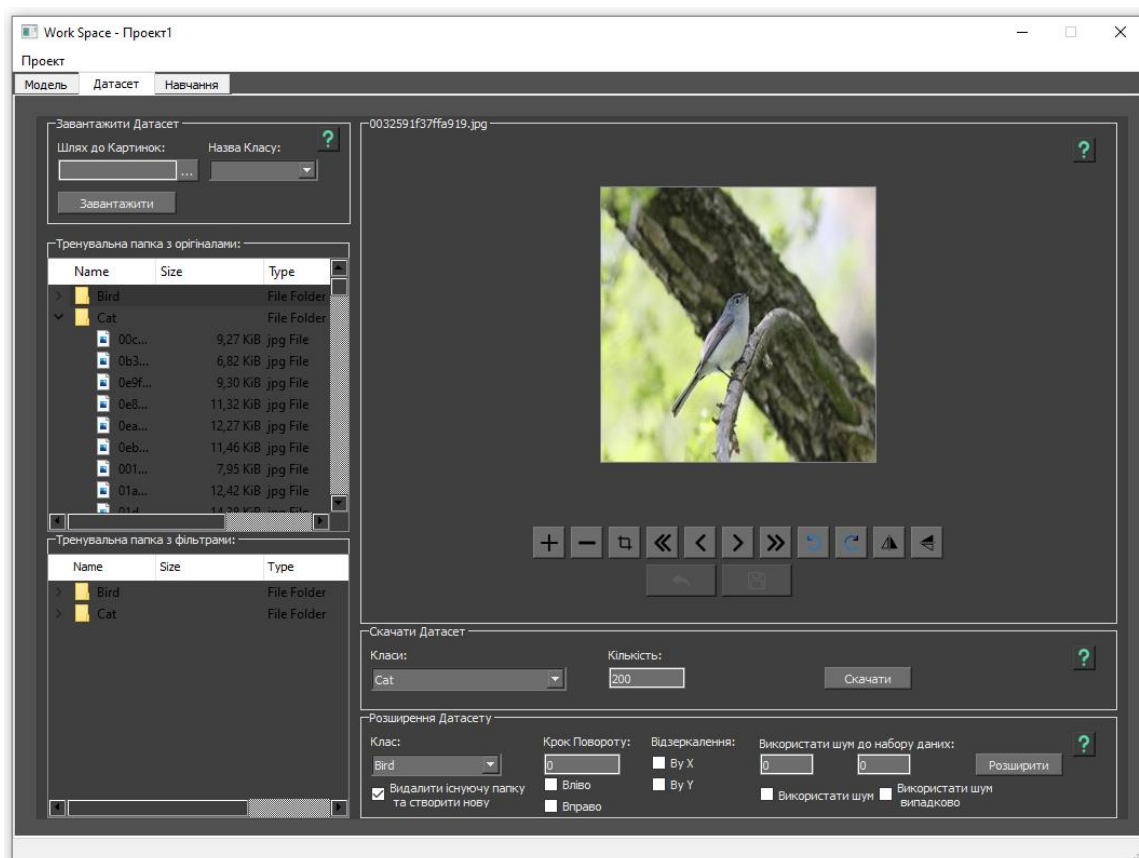


Рисунок 4.9 – Робота з окремою картинкою

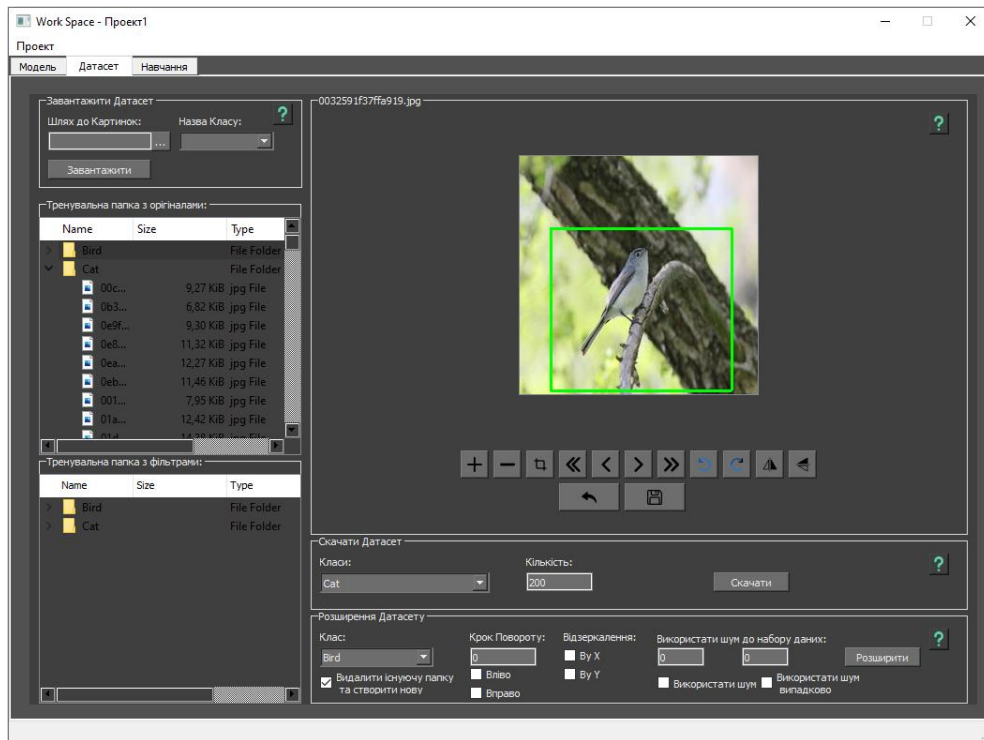


Рисунок 4.10 – Обрізання картинки

Після виконання всіх попередніх кроків, класи готові до розширення. Для цього користувачу необхідно вибрати випадальному списку "Клас" один із наявних класів, який він хоче розширити. Потім користувач має можливість видалити папку з класом та створити нову, це потрібно в разі, якщо клас потрібно розширити іншим способом. Далі користувач повинен вказати параметри розширення, такі як кут повороту у градусах та напрямок (вліво або вправо) повороту зображень, відзеркалення по горизонталі або вертикалі, а також використання шуму. Для шуму потрібно ввести відсоток шуму (у першому полі для введення) та відсоток зображень, які будуть модифіковані (у другому полі для введення). У кінці користувач повинен натиснути кнопку "Розширити", після чого почнеться процес, який можна відстежувати за допомогою прогрес-бару у нижній частині. Приклад цього показано на рисунку 4.11.

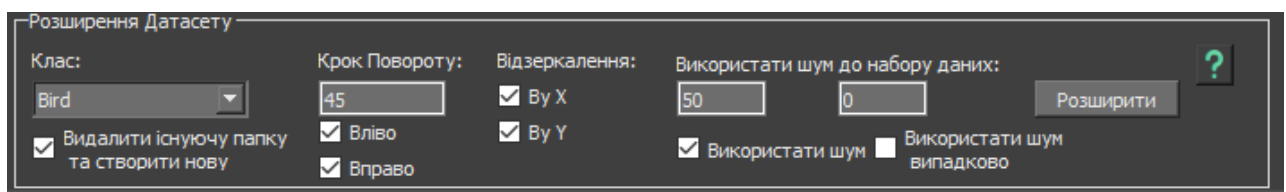


Рисунок 4.11 – Розширення класу

Після створення датасету, користувач повинен створити модель для навчання. Для цього він має перейти на вкладку "Навчання". При переході на цю вкладку, користувач побачить вікно конструктора з наступними елементами (рисунок 4.12):

- комбінований список: Цей елемент відповідає за пресети. Тут користувач може переглянути всі наявні пресети та ввести назву для нового пресету;
- дві кнопки керування пресетами: Перша кнопка з піктограмою корзинки видаляє пресет зі списку та з проекту. Друга кнопка з піктограмою плюса додає порожній пресет, що видаляє назву та очищає робочий простір від усіх елементів;
- кнопка "Підтвердити": Ця кнопка зберігає поточний стан робочого простору до вибраного пресету і генерує скрипт моделі, який буде використаний під час навчання;
- робочий простір: Тут відображаються всі елементи створеної моделі.
- зелена кнопка: Ця кнопка додає новий порожній елемент до робочого простору.

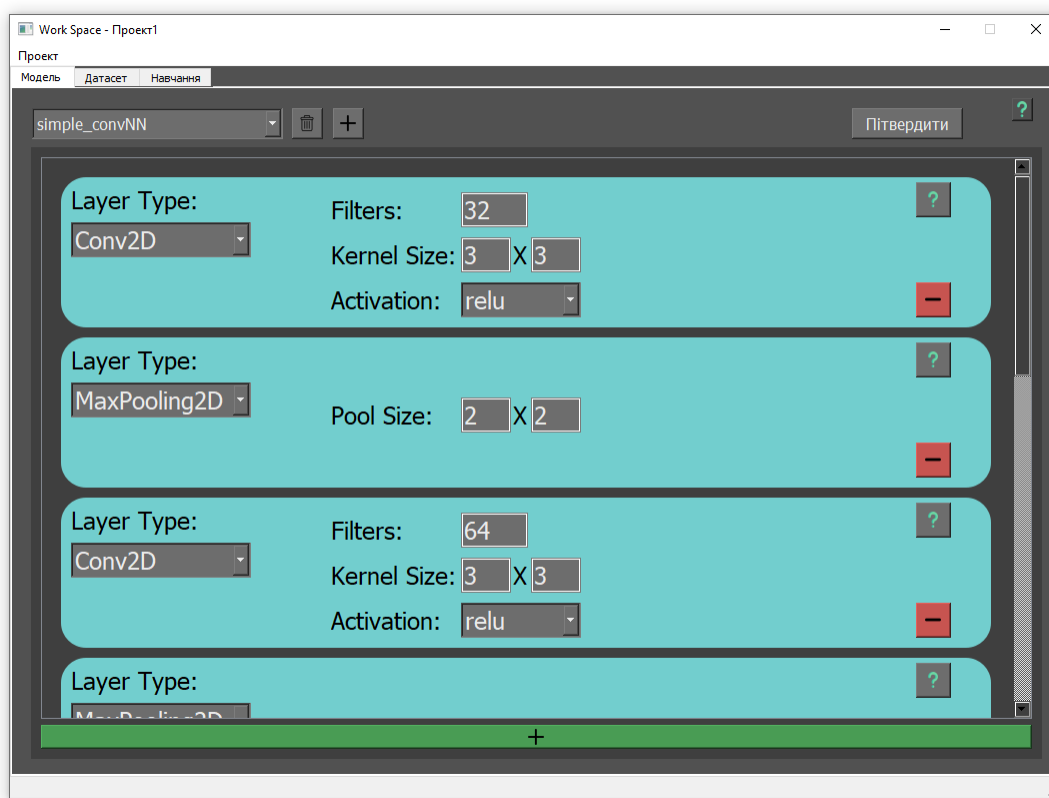


Рисунок 4.12 – Вікно конструктора моделей

Модель можна будувати, комбінуючи різні блоки, як показано на рисунку 4.12. Кожен блок складається з послідовних шарів, які формують нейронну мережу. Кожен шар має налаштовувані параметри, такі як кількість нейронів у шарі (Filters), розмір ядра (Kernel Size), Pool Size та активаційну функцію, яка застосовується до оброблених даних. Вибір параметрів також залежить від типу шару, таких як згортковий шар (Conv2D), шар прямого поширення (Dense), шар перетворення вхідних даних (Flatten) та шар MaxPooling2D.

Щоб перейти до процесу навчання, користувач повинен клацнути на вкладку "Навчання". Він побачить вікно для налаштування навчання та валідації нейронної мережі, як показано (рисунку 4.13). Це вікно містить наступні елементи:

– навчання: В цьому розділі користувач може налаштовувати параметри навчання. Він може вибрати кількість епох, кількість даних, які обробляються за кожну партію, оптимізатор, функцію втрат та можливість використовувати чекпоінти (збереження проміжних результатів);

– консоль (рисунок 4.14): Це інтегрована консоль, яка виводить прогрес навчання, такий як інформацію про кожну епоху та стан на кожній партії. Відображається значення втрат та відсоток правильних відповідей на тренувальних даних;

– графіки (рисунок 4.15): Після завершення навчання користувач може переглянути графіки, які відображають тенденції навчання на тренувальних та валідаційних даних протягом епох;

– валідація (рисунок 4.16): В цьому розділі користувач може завантажити навчену модель та протестувати її на власних зображеннях або на випадкових даних з датасету. Він отримує результат передбачення та відсоток точності.

					ІК93.018БАК.005 ПЗ	Арк.
Зм.	Лист	№ докум.	Підпис	Дата		60

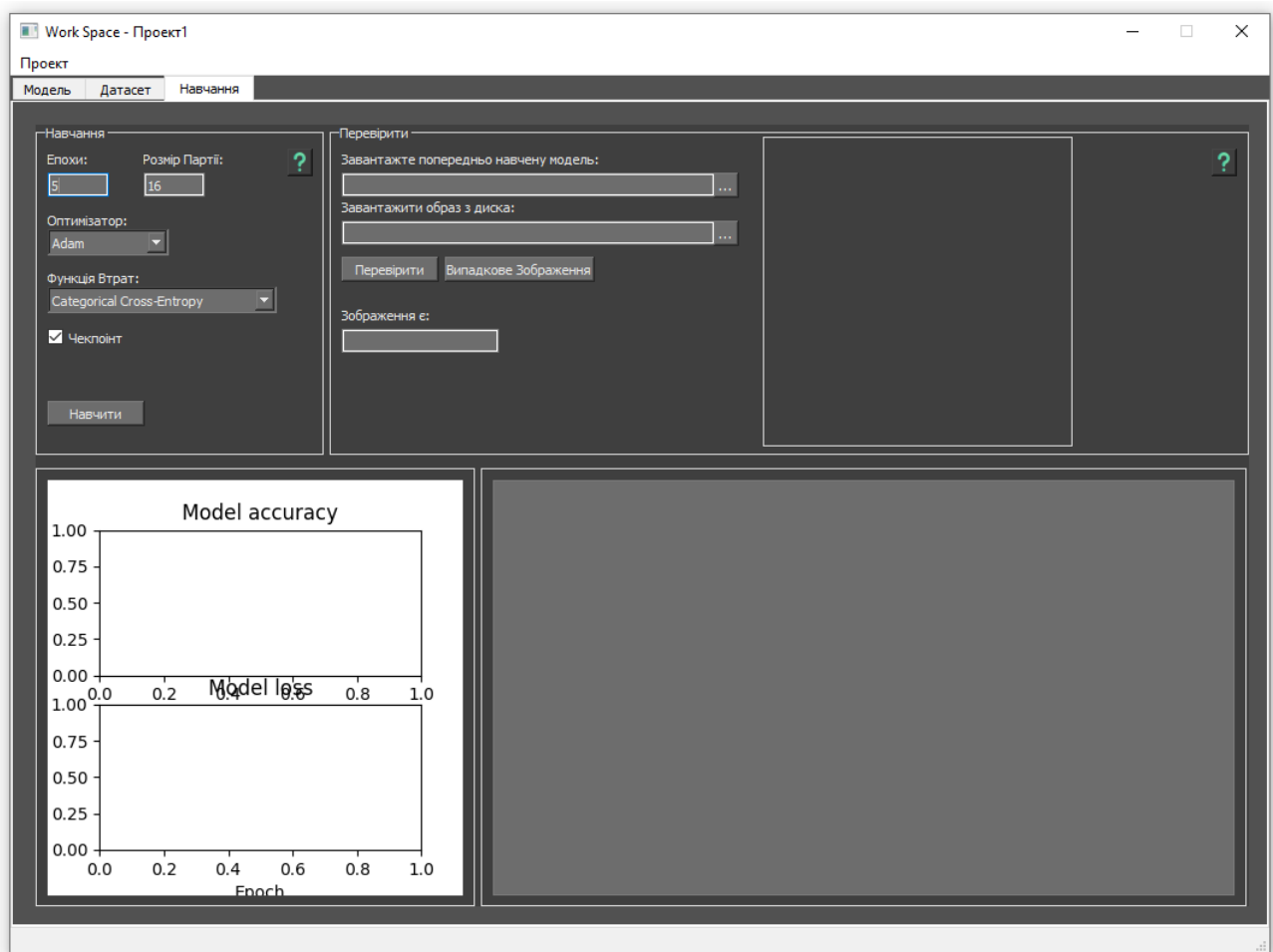


Рисунок 4.13 – Вікно навчання НМ

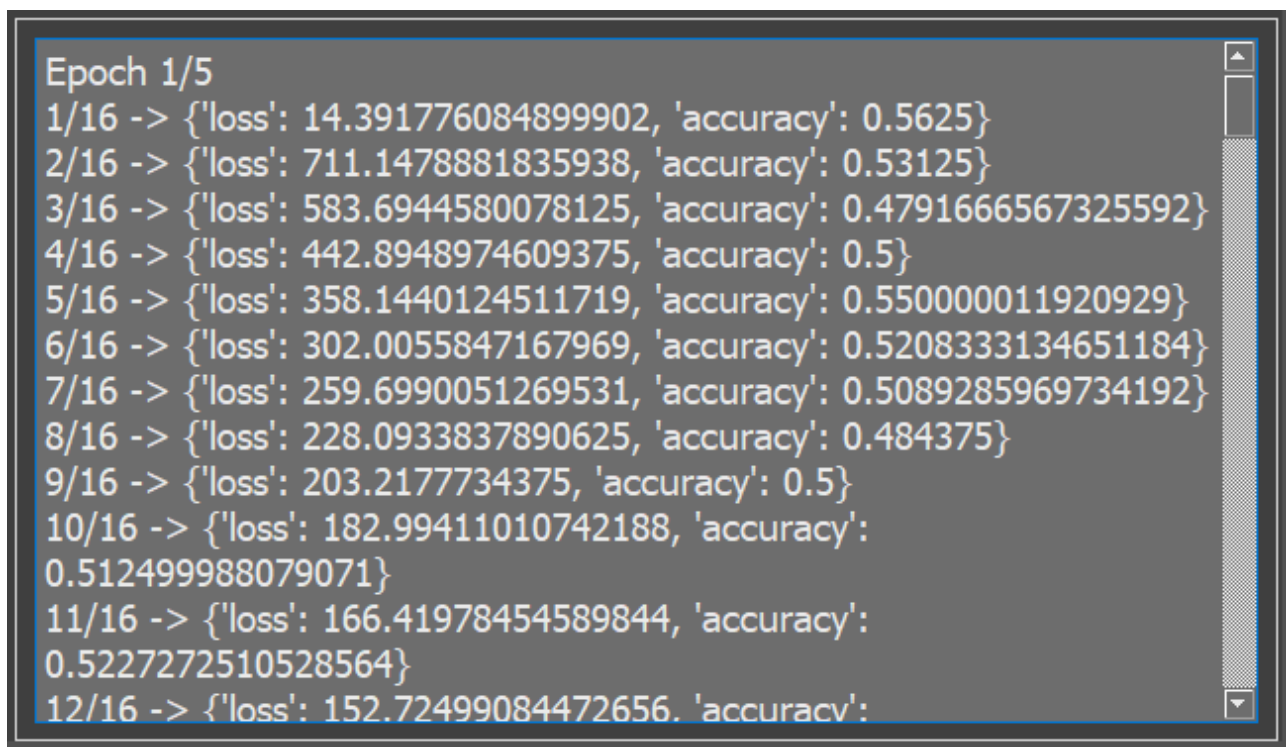


Рисунок 4.14 – Приклад навчання і результатів в консолі

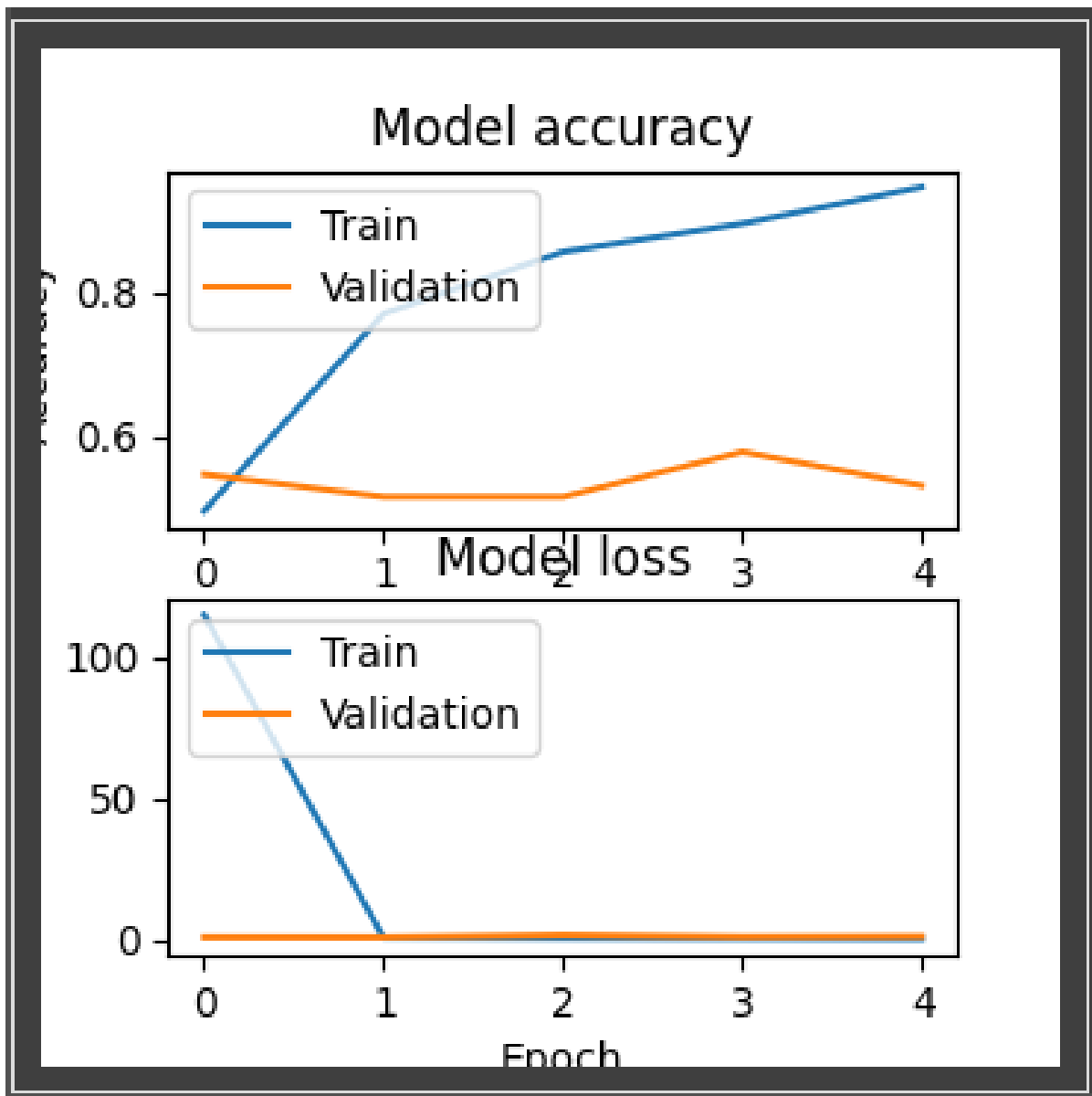


Рисунок 4.15 – Графіки після навчання



Рисунок 4.16 – Приклад валідації навченої моделі

## Висновки по розділу

У четвертому розділі ми детально розглянули покрокову роботу з нашим додатком. Ми надали приклади та докладно описали процес створення датасету для навчання, його обробку та розширення. Крім того, ми продемонстрували та пояснили процес створення моделі, який включав побудову графічно та створення скрипта для використання моделі в кодї. На завершення, ми показали процес навчання та валідації зображень.

Таким чином, ми описали повний життєвий цикл нашого додатку та надали детальну інструкцію для користувача.

					ІК93.018БАК.005 ПЗ	Арк.
Зм.	Лист	№ докум.	Підпис	Дата		63

## ВИСНОВКИ

Навчання нейронних мереж – це складний процес, який вимагає гарно підготовлених датасетів. Датасети виступають основою для навчання моделі і повинні бути репрезентативними, різноманітними та достатньо обсяжними, щоб модель могла зрозуміти та узагальнити закономірності в даних.

Підготовка гарних датасетів полягає в кількох важливих кроках. По-перше, потрібно зібрати достатню кількість даних, що відповідають області задачі, яку ми хочемо вирішити. Це може включати фотографії, відео, текст, аудіо або інші типи даних. По-друге, необхідно провести обробку даних, яка може включати розмітку, нормалізацію, вирівнювання, фільтрацію шуму та інші техніки для покращення якості даних і забезпечення однорідності.

Ціль даного проекту полягала у розробці унікального рішення, яке відрізняється від існуючих підходів та надає нові можливості для організації та автоматизації процесу створення датасетів для навчання нейронних мереж.

У ході роботи, було проведено аналіз Інтернету на наявність подібних рішень та прийнято рішення розробити десктоп-додаток, що допоможе:

- створити та підготувати датасет для навчання;
- зручно створювати та еспериментувати з моделями для даного датасету;
- швидко навчити та тут же перевірити якість навчання.

Як результат, було розроблено систему, що повністю задовольняє поставлені перед нею вимоги та задачі. Даний застосунок є дуже мобільною та зручною версією автоматизованого рішення, й може бути використаним будь-ким навіть якщо він не має навички програмування.

					ІК93.018БАК.005 ПЗ	Арк.
Зм.	Лист	№ докум.	Підпис	Дата		64

## ПЕРЕЛІК ПОСИЛАНЬ

1. Adrian Rosebrock. Deep Learning for Computer Vision with Python (2017).
2. Sandipan Dey. Hands-On Image Processing with Python (2018).
3. About LabelImg. URL: <https://github.com/heartexlabs/labelImg> (дата звернення: 05.05.2023).
4. About Nyckel. URL: <https://www.nyckel.com/about> (дата звернення: 05.05.2023).
5. About Viso Suite. URL: <https://viso.ai/company/> (дата звернення: 05.05.2023).
6. Indra den Bakker. Python Deep Learning Cookbook (2017).
7. Python for Computer Vision with OpenCV and Deep Learning. PyImageSearch. URL: <https://www.pyimagesearch.com/pyimagesearch-gurus/> (дата звернення: 05.05.2023).
8. Sam Newman. Building Microservices: Designing Fine-Grained Systems (2014).
9. About TensorFlow. URL: <https://www.tensorflow.org/> (дата звернення: 05.05.2023).
10. Aurélien Géron. Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow (2019).
11. About Keras. URL: <https://keras.io/api/> (дата звернення: 05.05.2023).
12. About OIDv4 ToolKit. URL: [https://github.com/EscVM/OIDv4\\_ToolKit](https://github.com/EscVM/OIDv4_ToolKit) (дата звернення: 25.05.2023).
13. About Open Image Dataset. URL: [https://storage.googleapis.com/openimages/web/factsfigures\\_v4.html](https://storage.googleapis.com/openimages/web/factsfigures_v4.html) (дата звернення: 25.05.2023).
14. About PIL. URL: <https://pillow.readthedocs.io/en/stable/> (дата звернення: 05.05.2023).
15. Guillaume Lazar, Robin Penea. Mastering Qt5 (2016).
16. About Qt. URL: <https://doc.qt.io/> (дата звернення: 05.05.2023).
17. François Chollet. Deep Learning with Python (2017).

					ІК93.018БАК.005 ПЗ	Арк.
Зм.	Лист	№ докум.	Підпис	Дата		65

18. Документація Python. URL: <https://docs.python.org/> (дата звернення: 05.05.2023).

19. About PyCharm. URL: <https://www.jetbrains.com/pycharm/learn/> (дата звернення: 05.05.2023).

					ІК93.018БАК.005 ПЗ	Арк.
Зм.	Лист	№ докум.	Підпис	Дата		66