**МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ**

**НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ «ЛЬВІВСЬКА ПОЛІТЕХНІКА»**

**ІКНІ**

Кафедра **ПЗ**



**ЗВІТ**

до лабораторної роботи №1

**на тему:** *“Створення та навчання моделі*

*комп'ютерного зору для класифікації*

*зображень”*

**з дисципліни** *“Штучний інтелект в ігрових застосунках”*

**Лектор:**

асис. каф. ПЗ

Бауск О. Є.

**Виконав:**

ст. гр. ПЗ-33

Юшкевич. А.І.

**Прийняв:**

ст. викл. каф. ПЗ

Бауск О. Є.

«\_\_» \_\_\_\_\_\_\_\_\_ 2024р.

∑=\_\_\_\_\_

Львів – 2024

**Тема роботи:** Створення та навчання моделі глибокого навчання для класифікації зображень з використанням PyTorch, Kaggle, Jupyter Notebook та FastAI.

**Мета роботи:** Ознайомитись з основами глибокого навчання, навчитися створювати набори даних для навчання моделей комп'ютерного зору, використовувати бібліотеку fastai для навчання моделей класифікації зображень, та розгортати навчені моделі.

# Теоретичні відомості

Jupyter Notebook - це інтерактивне середовище, яке дозволяє поєднувати код, текст, візуалізації та інші медіа в одному документі. Це ідеальний інструмент для експериментів з моделями машинного навчання, оскільки він дозволяє швидко змінювати та тестувати код.

Щоб почати роботу з Jupyter Notebook, необхідно встановити його за допомогою команди pip install jupyter та запустити за допомогою jupyter notebook. Це відкриє веб-інтерфейс, де ви зможете створювати та редагувати ноутбуки.

В контексті даної лабораторної роботи, Jupyter Notebook буде використовуватися для створення та навчання моделей глибокого навчання, але ми будемо використовувати Kaggle для цього.

Kaggle - це онлайн-платформа для машинного навчання, яка надає доступ до великого набору даних та моделей. Вона також включає в себе Jupyter Notebook, що дозволяє нам створювати та навчати моделі глибокого навчання. Таким чином, ми можемо використовувати Jupyter Notebook для створення та навчання моделей глибокого навчання, не встановлюючи його на своєму комп'ютері, і взагалі не встановлювати нічого на локальній машині.

Навчання моделі починається з підготовки даних. Наприклад, для класифікації зображень ми можемо використовувати набір даних з фотографіями котів та собак. Використовуючи бібліотеку fastai, ми можемо швидко створити модель, яка навчиться розпізнавати ці зображення.

Процес навчання включає кілька етапів: підготовка даних, створення моделі, навчання та оцінка якості. Fastai спрощує цей процес, надаючи інструменти для автоматичної обробки даних, налаштування моделі та моніторингу навчання.

Однією з ключових переваг глибокого навчання є можливість використання попередньо навчених моделей, що дозволяє значно скоротити час навчання та покращити якість результатів. Цей підхід, відомий як передавальне навчання, дозволяє адаптувати модель до нових задач, використовуючи вже наявні знання.

Глибоке навчання - це підрозділ машинного навчання, що базується на штучних нейронних мережах з багатьма шарами. Ці мережі здатні автоматично вивчати представлення даних з багатьма рівнями абстракції.

fastai - це бібліотека глибокого навчання високого рівня, побудована на основі PyTorch. Вона надає готові інструменти та найкращі практики для швидкого створення високоякісних моделей глибокого навчання.

# Завдання

1. Налаштувати середовище розробки в Kaggle.

2. Створити набір даних для класифікації зображень з двох класів.

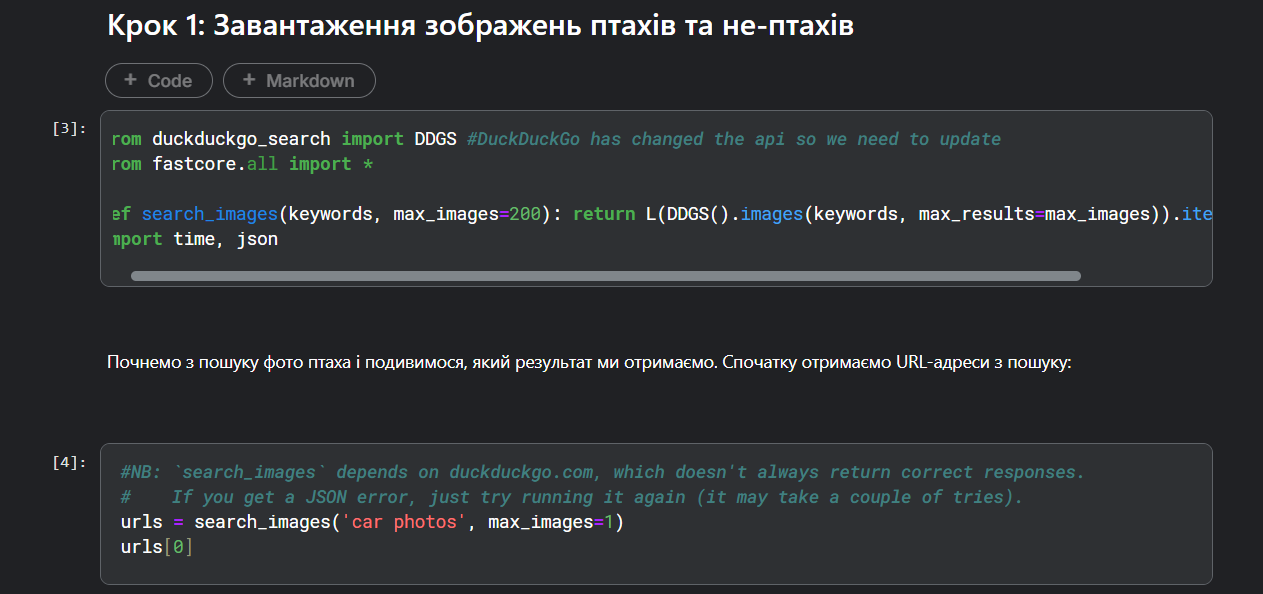
3. Створити модель-класифікатор.

4. Натренувати та використати модель для класифікації зображень.

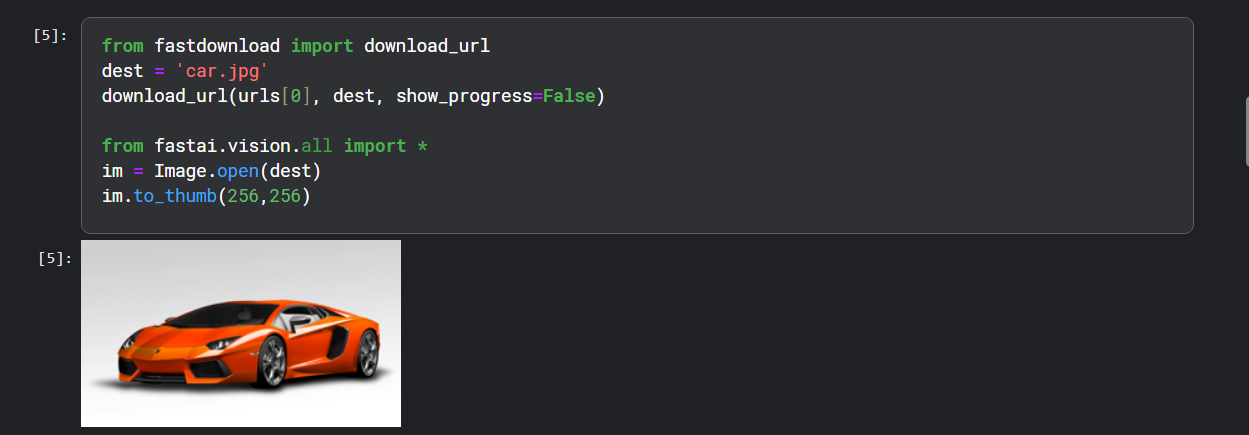
5. Побудувати власний класифікатор для інших класів зображень.

**Індивідуальний варіант**: 2. Автомобілі vs Мотоцикли

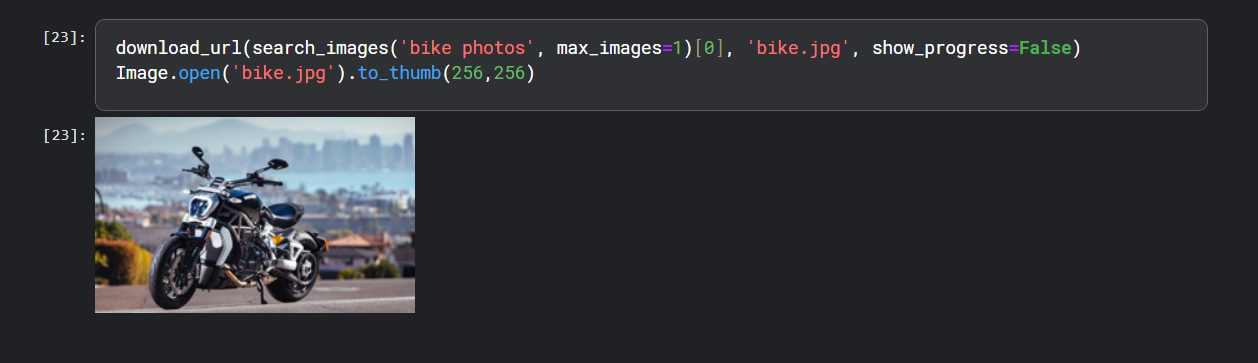
# Хід роботи

****

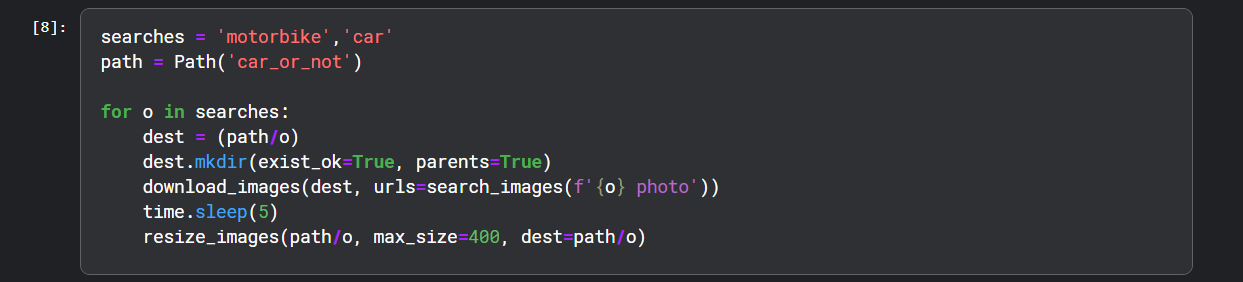
**­***Рис. 1. Отримання посилання на зображення автомобіля*

**

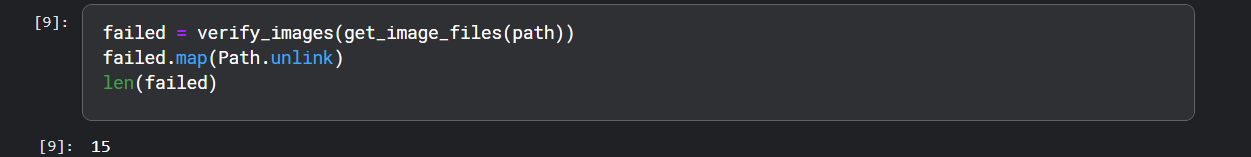
**­***Рис. 2. Завантаження зображення автомобіля*

**

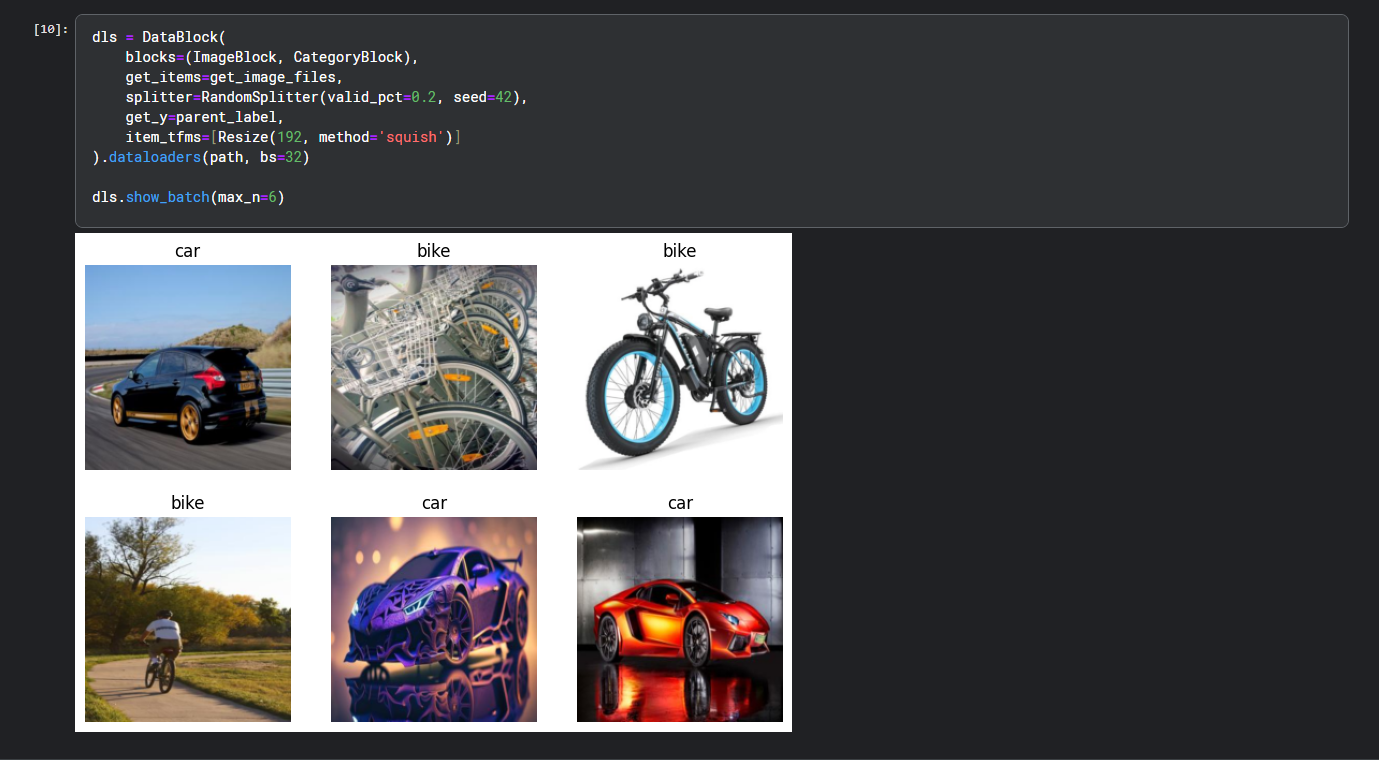
**­***Рис. 3. Пошук та завантаження зображення мотоцикла*

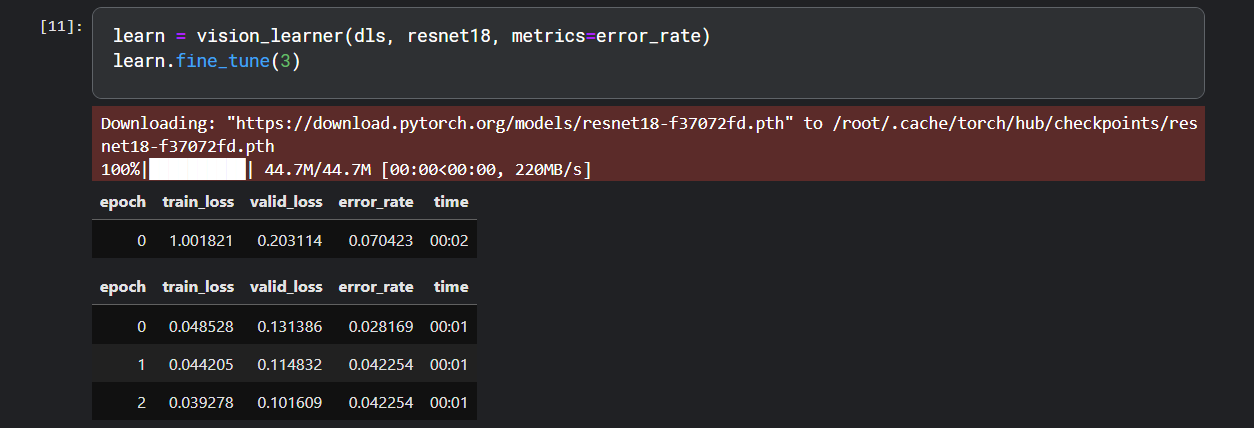
**

**­***Рис. 4. Запис набору зображень авто і мото в папку*

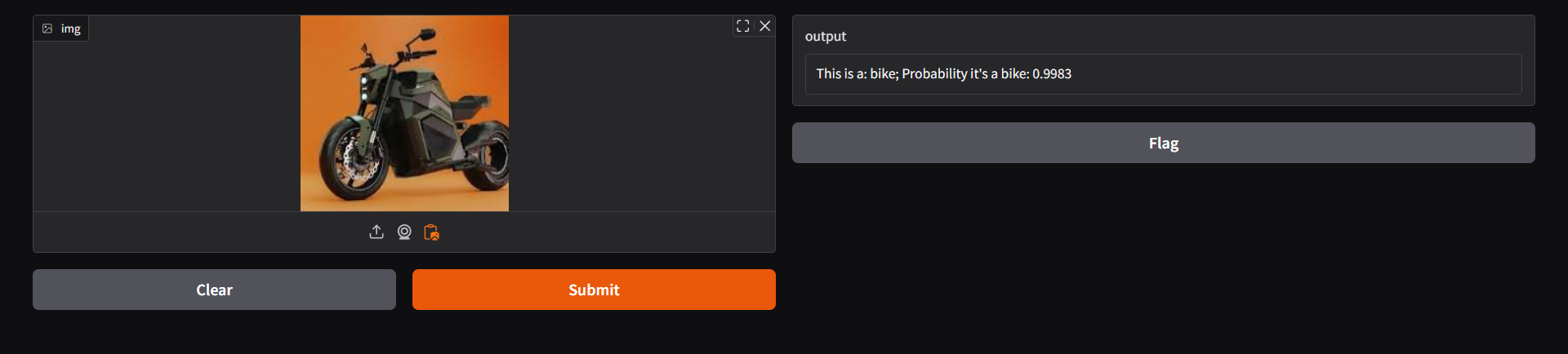
**

**­***Рис. 5. Верифікація зображень*

**

**­***Рис. 6. Створення тренувального сету  
  
*

**­***Рис. 7. Тренування*

*  
Рис. 8. Інтерфейс*

# Код інтерфейсу

import gradio as gr  
import pathlib  
import platform  
  
if platform.system() == 'Windows':  
 pathlib.PosixPath = pathlib.WindowsPath # Trick fastai into using WindowsPath  
  
from fastai.vision.all import load\_learner  
  
# Load the model  
learn = load\_learner("model.pkl")  
  
  
def classify\_image(img):  
 pred, \_, probs = learn.predict(img)  
 return f"This is a: {pred}; Probability it's a {pred}: {probs[(lambda x: 0 if x == "bike" else 0)(pred)]:.04f}"  
  
  
gr.Interface(fn=classify\_image,inputs=gr.Image(),outputs=gr.Text()).launch(share=True, debug=True)

**Висновок:** Під час виконання лабораторної роботи я ознайомився з основами deep seek`у, навчився створювати набори даних для навчання моделей комп'ютерного зору, використовувати бібліотеку fastai для навчання моделей класифікації зображень, та розгортати навчені моделі.