**Міністерство освіти і науки України**

**Національний університет харчових технологій**

Кафедра інформаційних технологій, штучного інтелекту та кібербезпеки

**ЛАБОРАТОРНА РОБОТА № 4**

з дисципліни «ГЛИБИННЕ НАВЧАННЯ»

на тему: «РОЗРОБКА НЕЙРОННОЇ МЕРЕЖІ ДЛЯ КЛАСИФІКАЦІЇ

ЗОБРАЖЕНЬ.»

Виконав: Студент I курсу  
групи КН-1-3М

Кучерявий М. В.

Перевірив:\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**Київ — 2025**

**Мета роботи**

навчитись розробляти нейронну мережу для задач класифікації зображень.

**Хід виконання роботи**

Створимо нейронну мережу для класифікації зображень. Використаємо наступні класи 'літак', 'автомобіль', 'птах', 'кішка', 'олень', 'собака', 'жаба', 'кінь', 'корабель', 'вантажівка'. Навчати модель будемо на основі cifar10. Створимо наступну программу:

from google.colab import drive

drive.mount('/content/drive')

import tensorflow as tf

from tensorflow.keras import layers, models

import matplotlib.pyplot as plt

import numpy as np

from PIL import Image

(x\_train, y\_train), (x\_test, y\_test) = tf.keras.datasets.cifar10.load\_data()

# Нормалізація

x\_train, x\_test = x\_train / 255.0, x\_test / 255.0

# Назви класів

class\_names = ['літак', 'автомобіль', 'птах', 'кішка', 'олень',

               'собака', 'жаба', 'кінь', 'корабель', 'вантажівка']

model = models.Sequential([

    layers.Conv2D(32, (3, 3), activation='relu', input\_shape=(32, 32, 3)),

    layers.MaxPooling2D((2, 2)),

    layers.Conv2D(64, (3, 3), activation='relu'),

    layers.MaxPooling2D((2, 2)),

    layers.Conv2D(64, (3, 3), activation='relu'),

    layers.Flatten(),

    layers.Dense(64, activation='relu'),

    layers.Dense(10)  # 10 класів

])

# КОМПІЛЯЦІЯ ТА НАВЧАННЯ

model.compile(optimizer='adam',

              loss=tf.keras.losses.SparseCategoricalCrossentropy(from\_logits=True),

              metrics=['accuracy'])

history = model.fit(x\_train, y\_train, epochs=10,

                    validation\_data=(x\_test, y\_test))

# Шлях до зображення

img\_path\_car = "/content/drive/MyDrive/testImages/car.jpg"

img\_path\_ship = "/content/drive/MyDrive/testImages/Titanic.jpg"

def classify\_image(img\_path, model, class\_names):

    # Завантаження та підготовка зображення

    img = Image.open(img\_path).convert('RGB')

    img = img.resize((32, 32))

    img\_array = np.array(img) / 255.0

    input\_image = np.expand\_dims(img\_array, axis=0)

    # Прогнозування

    probability\_model = tf.keras.Sequential([model, tf.keras.layers.Softmax()])

    prediction = probability\_model.predict(input\_image)

    predicted\_class = np.argmax(prediction)

    # Візуалізація

    plt.imshow(img)

    plt.title(f"Прогноз: {class\_names[predicted\_class]}")

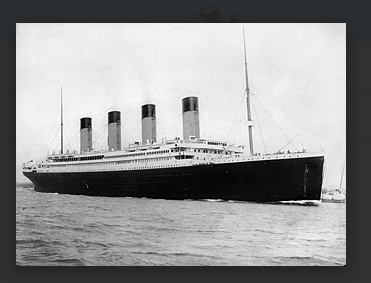
    plt.axis('off')

    plt.show()

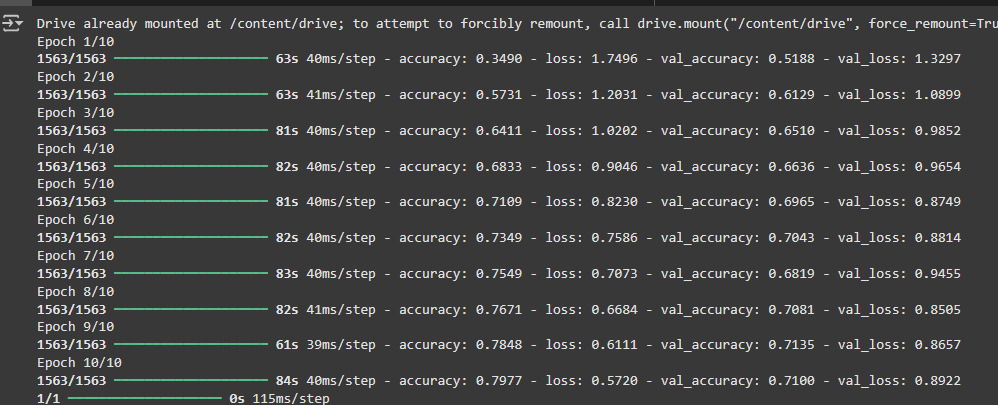
classify\_image(img\_path\_car, model, class\_names)

classify\_image(img\_path\_ship, model, class\_names)

Тестувальні зображення



Результати навчання та роботи моделі



****

**Для яких завдань можна використовувати згорткові нейронні мережі?**

Згорткові нейронні мережі використовуються переважно для обробки зображень і відео. Вони ефективно розпізнають об’єкти, класифікують зображення, виконують сегментацію та виявлення об’єктів. Також CNN застосовують у медичній діагностиці (наприклад, аналіз знімків МРТ), системах відеоспостереження та в автопілотах. Завдяки своїй структурі, вони добре виявляють просторові залежності в даних.

**Який основний принцип роботи згорткових нейронних мереж?**

Основний принцип роботи згорткових нейронних мереж полягає в автоматичному виділенні ознак із зображень за допомогою **згорткових фільтрів**. Ці фільтри "ковзають" по зображенню і виявляють локальні шаблони, такі як краї, кути чи текстури. Потім отримані ознаки проходять через шари активації та підвибірки, що зменшують розмірність і зберігають важливу інформацію. У результаті мережа навчається розпізнавати складні об’єкти на основі простих ознак.

**Які основні компоненти згорткових нейронних мереж?**

Основні компоненти згорткової нейронної мережі — це згорткові шари, шари активації, шари підвибірки та повнозв’язні шари. Згорткові шари виділяють ознаки зі зображення, а активація (наприклад, ReLU) вводить нелінійність. Пулінг зменшує розмірність і зберігає головну інформацію. Повнозв’язні шари на завершення виконують класифікацію на основі витягнутих ознак.

**Які переваги мають згорткові нейронні мережі порівняно з традиційними методами обробки зображень?**

Згорткові нейронні мережі автоматично навчаються виявляти важливі ознаки зображень, тоді як традиційні методи потребують ручного виділення ознак. Вони краще справляються зі складними структурами, варіаціями об'єктів і шумами. CNN добре масштабуються та забезпечують високу точність у задачах розпізнавання. Також вони ефективно використовують просторову інформацію завдяки локальній обробці пікселів.

**Як працюють згорткові шари у нейронних згорткових мережах?**

Згорткові шари працюють шляхом **застосування фільтрів (ядер)** до вхідного зображення. Кожен фільтр ковзає (згортується) по зображенню й обчислює добуток значень пікселів і ваг фільтра, утворюючи **карту ознак**. Таким чином мережа виявляє локальні шаблони, як-от краї, кути чи текстури. Кожен фільтр навчається автоматично, щоб розпізнавати різні важливі елементи зображення.