**МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ ФЕДЕРАЦИИ**

Федеральное государственное бюджетное образовательное учреждение высшего образования

САМАРСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ

Институт автоматики и информационных технологий

Кафедра «Информатика и Вычислительная техника»

**ОТЧЕТ**

**о выполнении лабораторной работы №7**

по дисциплине Компьютерные средства искусственного интеллекта

на тему Нейросеть с библиотеками

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Преподаватель** |  |  |  | А.А. Тюгашев |
|  | (должность) | (подпись) | (дата) | (инициалы, фамилия)  К.В. Портнов |
|  | (должность) | (подпись) | (дата) | (инициалы, фамилия) |
| **Студенты** | 4-ИАИТ-119 |  |  | Е.А. Щаев |
|  | (группа) | (подпись) | (дата) | (инициалы, фамилия) |

Самара 2024 г.

**Цель работы.** Разработать программу для классификации изображений, используя обученную модель MobileNetV2.

**Программный код на языке Python:**

import tensorflow as tf

from tensorflow.keras.applications import MobileNetV2

from tensorflow.keras.applications.mobilenet\_v2 import preprocess\_input, decode\_predictions

from tensorflow.keras.preprocessing import image

import numpy as np

import matplotlib.pyplot as plt

import os

# Загрузка обученной модели

model = MobileNetV2(weights='imagenet')  # Загружаем модель с весами ImageNet

# Функция для загрузки и подготовки изображений

def load\_and\_preprocess\_images(image\_paths):

    images = []

    for img\_path in image\_paths:

        # Загрузка изображения и изменение размера

        img = image.load\_img(img\_path, target\_size=(224, 224))

        img\_array = image.img\_to\_array(img)  # Преобразуем в массив NumPy

        img\_array = preprocess\_input(img\_array)  # Предобработка

        images.append(img\_array)

    return np.array(images)

def display\_predictions(image\_paths, predictions):

    for img\_path, prediction in zip(image\_paths, predictions):

        # Отображение изображения

        img = image.load\_img(img\_path, target\_size=(224, 224))

        plt.imshow(img)

        plt.axis('off')

        plt.title("Результаты классификации")

        plt.show()

        # Проверяем размерность предсказаний

        if len(prediction.shape) == 1:  # Если одномерный массив

            prediction = np.expand\_dims(prediction, axis=0)  # Преобразуем в 2D-формат

        # Топ-3 предсказания

        decoded\_predictions = decode\_predictions(prediction, top=3)

        print(f"Предсказания для {os.path.basename(img\_path)}:")

        for i, (imagenet\_id, label, score) in enumerate(decoded\_predictions[0]):

            print(f"  {i+1}: {label} ({score:.2f})")

        print()

# Указываем пути к изображениям

image\_paths = []

# Загрузка и предобработка изображений

images = load\_and\_preprocess\_images(image\_paths)

# Прогнозирование классов

predictions = model.predict(images)

# Отображение результатов

display\_predictions(image\_paths, predictions)

**Пример работы программы:**

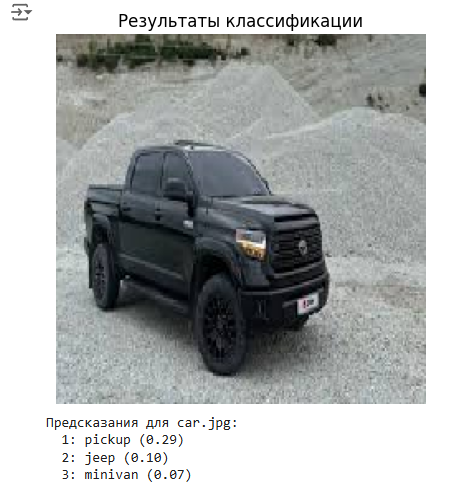
****

Рисунок 1 – Нейросеть определила пикап



Рисунок 2 – Нейросеть определила Африканского слона

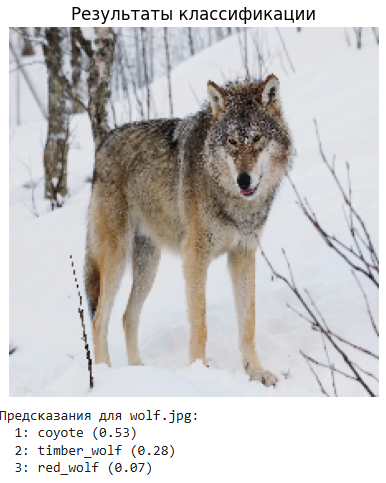


Рисунок 3 – Нейросеть определила койота

**Вывод:** Проверили работу обученной модели на классификацию изображений.