



**МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ
РОССИЙСКОЙ ФЕДЕРАЦИИ**
Федеральное государственное бюджетное образовательное учреждение
высшего образования
САМАРСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ
Институт автоматизации 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

Результаты классификации



Предсказания для wolf.jpg:

- 1: coyote (0.53)
- 2: timber_wolf (0.28)
- 3: red_wolf (0.07)

Рисунок 3 – Нейросеть определила койота

Вывод: Проверили работу обученной модели на классификацию изображений.