Минобрнауки России

Федеральное государственное бюджетное образовательное

учреждение высшего образования

**НИЖЕГОРОДСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ**

**ИМ. Р.Е. АЛЕКСЕЕВА**

ИНСТИТУТ РАДИОЭЛЕКТРОНИКИ И ИНФОРМАЦИОННЫХ ТЕХНОЛОГИЙ

|  |  |  |
| --- | --- | --- |
|  |  |  |

Курс “Аппаратное и программное обеспечение роботизированных систем”

**Отчет по лабораторной работе №3**

Выполнили:

Михеев И. С

Проверил:

Гай В.Е.

Нижний Новгород 2021

Задание:

Используя модуль Keras, написать алгоритм работы нейронной сети для распознавания изображений.

Ход работы:

Для начала мы подключаем все модели, которые будут необходимы для реализации нейронной сети:

import keras

import tensorflow as tf

from tensorflow.python.keras.preprocessing.image import ImageDataGenerator

from keras.applications.mobilenet import MobileNet, preprocess\_input

from keras.models import Sequential

from keras.layers.core import Flatten, Dense, Dropout

from google.colab import drive

Затем нам необходимо подключиться к гугл диску, чтобы занести туда файлы, на которых будет тренироваться наша нейронная сеть, а также тестовую выборку:

drive.mount('/content/drive')

train\_dir = '/content/drive/MyDrive/Keras/train'

test\_dir = '/content/drive/MyDrive/Keras/validation'

Дальше соответственно задаем размер изображения, размер мини-выборки, количество изображений для обучения и для теста. Все эти значения нам понадобятся в дальнейшем:

img\_width, img\_height = 500, 500

batch\_size = 79

nb\_train\_samples = 1034

nb\_test\_samples = 133

При помощи ImageDataGenerator создаем генератор изображений:

datagen = ImageDataGenerator(rescale=1. / 255)

train\_generator = datagen.flow\_from\_directory(

    train\_dir,

    target\_size=(img\_width, img\_height),

    batch\_size=batch\_size,

    class\_mode='categorical')

test\_generator = datagen.flow\_from\_directory(

    test\_dir,

    target\_size=(img\_width, img\_height),

    batch\_size=batch\_size,

    class\_mode='categorical')

Создаем экземпляр модели сети MobileNet

Mobile\_Net = MobileNet(

    weights='imagenet',

    include\_top=False,

    input\_shape=(img\_width, img\_height, 3)

)

Mobile\_Net.trainable = True

trainable = False

for layer in Mobile\_Net.layers:

  if layer.name == 'conv\_dw\_13':

    trainable = True

  layer.trainable = trainable

Mobile\_Net.summary()

Создаем составную сеть:

model = Sequential()

model.add(Mobile\_Net)

model.add(Flatten())

model.add(Dense(256, activation='relu'))

model.add(Dropout(0.5))

model.add(Dense(3, activation='sigmoid'))

model.summary()

Компилируем и обучаем сеть

model.compile(loss='categorical\_crossentropy',

              optimizer='sgd',

              metrics=['accuracy'])

model.fit\_generator(

    train\_generator,

    steps\_per\_epoch=nb\_train\_samples // batch\_size,

    epochs=10,

    validation\_data=test\_generator,

    validation\_steps=nb\_test\_samples // batch\_size)

Результат

