МИНИСТЕРСТВО науки и высшего ОБРАЗОВАНИЯ РОссИЙСКОЙ ФЕДЕРАЦИИ

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ

УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ

«МОСКОВСКИЙ АВИАЦИОННЫЙ ИНСТИТУТ

(национальный исследовательский университет)»

МАИ

Институт №3 «Системы управления, информатика и электроэнергетика»

Кафедра № 304 «Вычислительные машины, системы и сети»

**Экспертные системы**

Отчет по лабораторной работе №5-6

“Распознавание фруктов при помощи OpenCV”

Выполнили студенты группы М3О-410Б-21:

Дудоров Д.А. \_\_\_\_\_\_\_\_\_

Шилин И.С. \_\_\_\_\_\_\_\_\_

Приняли:

Трембач В.М. \_\_\_\_\_\_\_\_\_

**Описание задачи**

Разработать и обучить свёрточную нейронную сеть для классификации изображений фруктов из набора данных Fruits-360. Провести предварительную обработку данных и аугментацию изображений, после чего оценить качество обученной модели на тестовых данных.

**Теория**

Нейронные сети представляют собой математические модели, имитирующие структуру и функционирование человеческого мозга. Они состоят из нейронов, организованных в слои, где каждый нейрон выполняет операцию взвешенного суммирования входных данных с последующим применением функции активации.

Свёрточные нейронные сети (CNN) – это специализированные нейронные сети, предназначенные для анализа изображений. В основе работы CNN лежат свёрточные слои, которые используют фильтры (ядра свёртки), позволяющие выделять значимые признаки из входных данных, такие как границы, текстуры и различные паттерны.

Ключевые компоненты CNN:

• Свёрточный слой (Conv2D) – слой, применяющий свёрточные фильтры к изображениям.

• Pooling слой (MaxPooling) – снижает размерность изображений, сохраняя значимые признаки.

• Flatten слой – преобразует двумерные данные в одномерный вектор признаков.

• Полносвязные слои (Dense) – используют выделенные признаки для окончательной классификации.

• Функции активации (ReLU, softmax) – вводят нелинейность и выполняют классификацию на конечном этапе.

**Реализация программы**

from keras.models import Sequential

from keras.layers import Conv2D, MaxPooling2D, Activation, Dropout, Flatten, Dense

from tensorflow.keras.preprocessing.image import ImageDataGenerator, img\_to\_array, load\_img

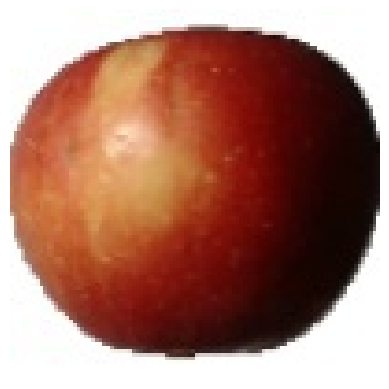
import matplotlib.pyplot as plt

import warnings

warnings.filterwarnings('ignore')

train\_path = "/input/fruits/fruits-360\_100x100/fruits-360/Training/"

test\_path = "/input/fruits/fruits-360\_100x100/fruits-360/Test/"



**Обучение модели**

img = load\_img(train\_path + "Apple Braeburn 1/0\_100.jpg")

model = Sequential()

model.add(Conv2D(32,(3,3), input\_shape = img.shape))

model.add(Activation("relu"))

model.add(MaxPooling2D())

model.add(Conv2D(32,(3,3)))

model.add(Activation("relu"))

model.add(MaxPooling2D())

model.add(Conv2D(64,(3,3)))

model.add(Activation("relu"))

model.add(MaxPooling2D())

model.add(Flatten())

model.add(Dense(1024))

model.add(Activation("relu"))

model.add(Dropout(0.5))

model.add(Dense(numberOfClass))

model.add(Activation("softmax"))

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

optimizer = "rmsprop",

metrics = ["accuracy"])

batch\_size = 32

train\_datagen = ImageDataGenerator(rescale = 1./255,

shear\_range = 0.3,

horizontal\_flip = True,

zoom\_range = 0.3)

test\_datagen = ImageDataGenerator(rescale = 1./255)

train\_generator = train\_datagen.flow\_from\_directory(train\_path,

target\_size = img.shape[:2],

batch\_size = batch\_size,

color\_mode = "rgb",

class\_mode = "categorical")

test\_generator = test\_datagen.flow\_from\_directory(train\_path,

target\_size = img.shape[:2],

batch\_size = batch\_size,

color\_mode = "rgb",

class\_mode = "categorical")

hist = model.fit(

train\_generator,

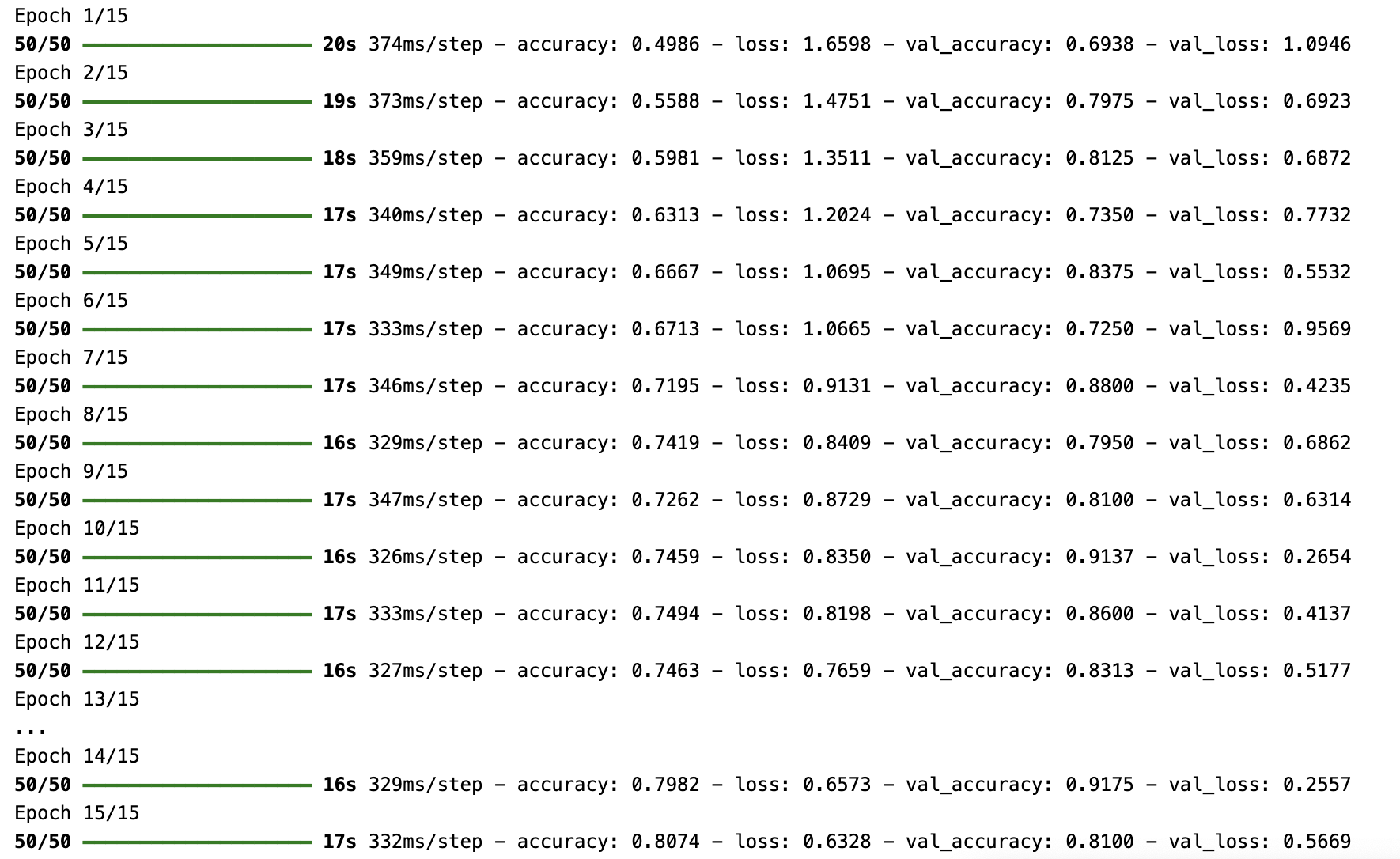
steps\_per\_epoch=1600 // batch\_size,

epochs=15,

validation\_data=test\_generator,

validation\_steps=800 // batch\_size)

model.save\_weights("mark.weights.h5")



**Метрики модели**

print(hist.history.keys())

plt.plot(hist.history["loss"], label = "Train Loss")

plt.plot(hist.history["val\_loss"], label = "Validation Loss")

plt.legend()

plt.show()

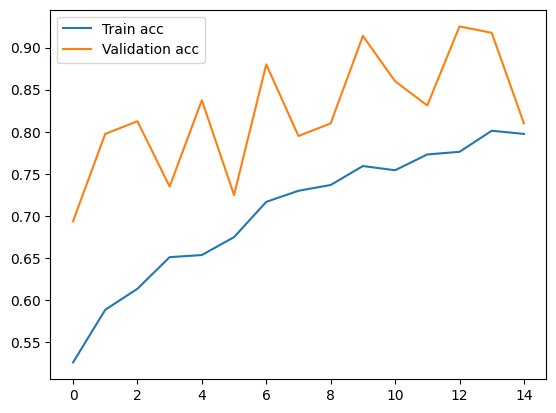
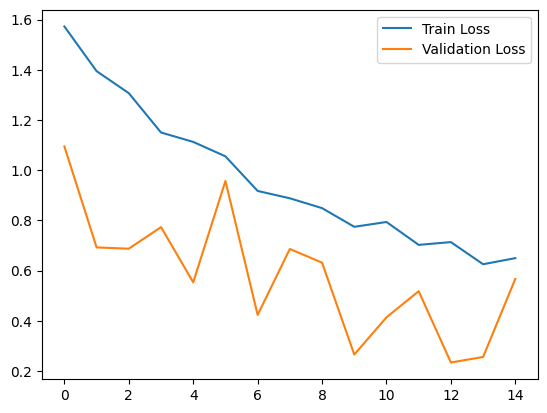
plt.figure()

plt.plot(hist.history["accuracy"], label="Train acc")

plt.plot(hist.history["val\_accuracy"], label="Validation acc")

plt.legend()

plt.show()



**Тестирование модели**

from tensorflow.keras.models import Sequential

from tensorflow.keras.layers import Conv2D, MaxPooling2D, Activation, Dropout, Flatten, Dense

from tensorflow.keras.preprocessing.image import load\_img, img\_to\_array

import numpy as np

model = Sequential()

model.add(Conv2D(32, (3, 3), input\_shape=(100, 100, 3)))

model.add(Activation("relu"))

model.add(MaxPooling2D())

model.add(Conv2D(32, (3, 3)))

model.add(Activation("relu"))

model.add(MaxPooling2D())

model.add(Conv2D(64, (3, 3)))

model.add(Activation("relu"))

model.add(MaxPooling2D())

model.add(Flatten())

model.add(Dense(1024))

model.add(Activation("relu"))

model.add(Dropout(0.5))

model.add(Dense(numberOfClass))

model.add(Activation("softmax"))

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

optimizer="rmsprop",

metrics=["accuracy"])

model.load\_weights("/working/mark.weights.h5")

img\_path = '/input/fruits/fruits-360\_100x100/fruits-360/Training/Apple 8/r0\_122\_100.jpg'

img = load\_img(img\_path, target\_size=(100, 100))

img\_array = img\_to\_array(img) / 255.0

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

predictions = model.predict(img\_array)

predicted\_class\_index = np.argmax(predictions)

predicted\_class\_label = list(train\_generator.class\_indices.keys())[predicted\_class\_index]

print(f'Предсказанный класс: {predicted\_class\_label}')

**Результат**

Предсказанный класс: Apple 10

**Вывод**

В рамках данной лабораторной работы были подробно изучены теоретические основы и практическое применение свёрточных нейронных сетей для решения задач классификации изображений. Разработанная модель успешно справилась с поставленной задачей, демонстрируя высокую точность. Приобретённые знания и навыки могут быть использованы при выполнении более сложных задач в области искусственного интеллекта и компьютерного зрения.