

МИНОБРНАУКИ РОССИИ  
Федеральное государственное бюджетное образовательное  
учреждение высшего образования  
«Ярославский государственный университет им. П.Г. Демидова»

Кафедра дискретного анализа

«Допустить к защите»

Заведующий кафедрой,

д.ф.-м.н., профессор

\_\_\_\_\_ Бондаренко В.А.

« 13 » июня 2020 г.

***Выпускная квалификационная работа бакалавра***  
по направлению 01.03.02 Прикладная математика и информатика

**Анализ алгоритмов глубокого машинного обучения в задачах  
распознавания изображений**

Научный руководитель

к.т.н., старший преподаватель

\_\_\_\_\_ Д.В. Матвеев

«    » \_\_\_\_\_ 2020 г.

Студент группы ИВТ-41БО

\_\_\_\_\_ А.С. Коротков

«    » \_\_\_\_\_ 2020 г.

Ярославль, 2020

## РЕФЕРАТ

Объем 17 стр., 6 гл., 6 источников

**Ключевые слова:** нейронные сети, глубокое машинное обучение, распознавание изображений, TensorFlow, OpenCV, Keras.

Объектом исследования являются методы на основе глубоких нейронных сетей для задач распознавания изображений.

Цель работы – разработка нейронной сети для решения задачи распознавания изображений.

В результате работы была разработана и реализована нейронная сеть для решения поставленной задачи. Проведен анализ полученных результатов и сделан вывод о качестве работы нейронных сетей в задачах распознавания изображений.

## СОДЕРЖАНИЕ

ВВЕДЕНИЕ . . . . .	4
1 ОБЗОР ПРЕДМЕТНОЙ ОБЛАСТИ . . . . .	6
1.1 Задачи компьютерного зрения . . . . .	6
1.2 Искусственные нейронные сети . . . . .	6
1.2.1 Понятие искусственной нейронной сети . . . . .	6
1.2.2 Архитектура нейронных сетей . . . . .	6
1.2.3 Глубокие нейронные сети . . . . .	6
1.2.4 Проблемы обучения нейронных сетей . . . . .	6
1.3 Применение глубоких нейронных сетей в задачах распознавания изображений . . . . .	6
2 ПОСТАНОВКА ЗАДАЧИ . . . . .	7
3 ПРОЕКТИРОВАНИЕ СИСТЕМЫ ДЛЯ ЗАДАЧА . . . . .	8
4 ПРАКТИЧЕСКАЯ РЕАЛИЗАЦИЯ И ПРИМЕНЕНИЕ ЗАДАЧА НА ИЗОБРАЖЕНИЯХ . . . . .	9
5 ПРИМЕРЫ РАБОТЫ ПРОГРАММЫ . . . . .	10
6 АНАЛИЗ РАБОТЫ ПРОГРАММЫ . . . . .	11
ЗАКЛЮЧЕНИЕ . . . . .	12
СПИСОК ЛИТЕРАТУРЫ . . . . .	13
СПИСОК ИЛЛЮСТРАТИВНОГО МАТЕРИАЛА . . . . .	14
ПРИЛОЖЕНИЕ А . . . . .	15

## ВВЕДЕНИЕ

В настоящее время, в связи со стремительным развитием цифровых технологий, использование автоматизированных и роботизированных систем распространилось на множество областей как в промышленности, науке, так и в повседневной жизни. В следствие этого, возрастает необходимость в эффективной обработке информации, представленной, в частности, в формате видео и изображений.

На текущий момент изображения тесно влились в жизнь человека. Поэтому многие автоматизированные системы используют их в качестве основного источника информации. Нахождение, локализация, классификация и анализ образов на изображении компьютером – сложная задача компьютерного зрения.

Компьютерное (машинное) зрение – это совокупность программно-технических решений в сфере искусственного интеллекта (ИИ), нацеленных на считывание и обработку изображений, в реальном времени и без участия человека.

В настоящий момент, такие технологии применяются для решения таких сложных задач как:

- OCR – Optical character recognition (Оптическое распознавание символов): преобразование текста на изображении в редактируемый.
- Фотограмметрия – технология создания трехмерной модели объекта на основе фотографий, сделанных с различных ракурсов.
- Motion capture – технология, широко применяемая в киноиндустрии, позволяющая преобразовывать движения реальных людей в компьютерную анимацию.
- Дополненная реальность (AR) – технология, позволяющая в реальном времени проецировать виртуальные объекты на изображение реального окружения.
- Медицинская диагностика – обнаружение раковых клеток на ранней стадии, увеличение качества МРТ изображений, их анализ и т.д.

В процессе обработки информации, получаемой из глаз, человеческий мозг پردازывает колоссальный объем работы. Человек без труда сможет описать что находится и что происходит на случайно взятой фотографии. Изображения могут нести в себе колоссальное количество деталей и отличаться множеством параметров, таких как: разрешение, цветность, качество, яркость, наличие шума и т.д. Объекты на изображениях также могут обладать множеством особенностей: масштаб, положение, цвет, поворот, наклон и т.д. Однако, в цифровом формате, каждое изображение представляет собой лишь

массив числовых данных. Научить компьютер находить и классифицировать образы на изображении с учетом всех факторов – очень сложная алгоритмическая задача. Для её решения активно применяют технологии машинного обучения.

В данной работе был проведен анализ алгоритмов глубокого машинного обучения для решения задач распознавания изображений, а также разработана система для ЗАДАЧА.

В первой главе ...

Во второй главе ...

В третьей главе ...

## 1 ОБЗОР ПРЕДМЕТНОЙ ОБЛАСТИ

### 1.1 Задачи компьютерного зрения

### 1.2 Искусственные нейронные сети

#### 1.2.1 Понятие искусственной нейронной сети

Машинное обучение – раздел исследований в сфере ИИ, в основе которых лежат методы разработки систем способных к обучению.

Искусственная нейронная сеть (ИНС) – компьютерная модель, в основе которой лежит упрощенное представление человеческого мозга. Структурной единицей ИНС является нейрон.

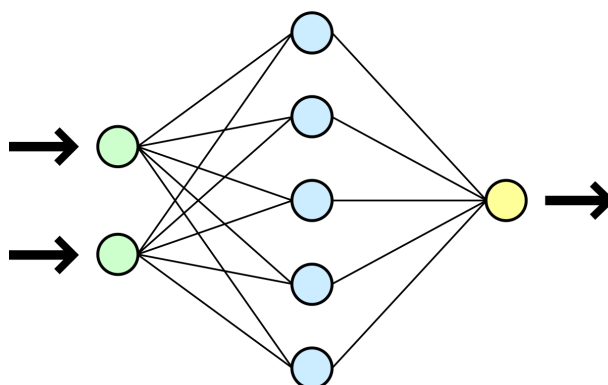


Рисунок 1.1 – Схема простой нейронной сети

#### 1.2.2 Архитектура нейронных сетей

#### 1.2.3 Глубокие нейронные сети

#### 1.2.4 Проблемы обучения нейронных сетей

### 1.3 Применение глубоких нейронных сетей в задачах распознавания изображений

## 2 ПОСТАНОВКА ЗАДАЧИ

### 3 ПРОЕКТИРОВАНИЕ СИСТЕМЫ ДЛЯ ЗАДАЧА



#### 4 ПРАКТИЧЕСКАЯ РЕАЛИЗАЦИЯ И ПРИМЕНЕНИЕ ЗАДАЧА НА ИЗОБРАЖЕНИЯХ

## 5 ПРИМЕРЫ РАБОТЫ ПРОГРАММЫ

## 6 АНАЛИЗ РАБОТЫ ПРОГРАММЫ

## ЗАКЛЮЧЕНИЕ

## СПИСОК ЛИТЕРАТУРЫ

1. *Michelucci U.* Applied Deep Learning: A Case-Based Approach to Understanding Deep Neural Networks. — Apress, 2018. — С. 431.
2. *Singh P., Manure A.* Learn TensorFlow 2.0: Implement Machine Learning and Deep Learning Models with Python. — Apress, 2020. — С. 195.
3. *Жерон О.* Прикладное машинное обучение с помощью Scikit-Learn и TensorFlow. — ООО "Альфа-книга", 2018. — С. 688.
4. *Конец Д.* Классические задачи Computer Science на языке Python. — Питер, 2020. — С. 256.
5. *Нишант Ш.* Машинное обучение и TensorFlow. — Питер, 2019. — С. 336.
6. *Шолле Ф.* Глубокое обучение на Python. — Питер, 2018. — С. 400.

## СПИСОК ИЛЛЮСТРАТИВНОГО МАТЕРИАЛА

Рисунок 1.1	Схема простой нейронной сети . . . . .	6
-------------	--	---

## ПРИЛОЖЕНИЕ А

### Листинг 1

```
1 from __future__ import absolute_import, division, print_function,
   unicode_literals
2
3 import numpy as np
4 import cv2
5 from tensorflow.keras.callbacks import ModelCheckpoint
6 from tensorflow.keras.layers import Conv2D, Flatten, MaxPooling2D,
   Dense, Dropout
7 from tensorflow.keras.models import Sequential
8 from tensorflow.keras.preprocessing.image import ImageDataGenerator,
   img_to_array, load_img, array_to_img
9 import random, os, glob
10 import matplotlib.pyplot as plt
11
12 DIR_PATH = '/home/alexandr/dev/datasets/garbage-classification/
   garbage_classification/Garbage_classification'
13 img_list = glob.glob(os.path.join(DIR_PATH, '*/*.jpg'))
14
15
16 def load_data():
17     train=ImageDataGenerator(
18         horizontal_flip=True, vertical_flip=True, validation_split
19         =0.1, rescale=1./255,
20         shear_range=0.1,
21         zoom_range=0.1,
22         width_shift_range=0.1,
23         height_shift_range=0.1)
24     test=ImageDataGenerator(rescale=1/255, validation_split=0.1)
25     train_generator=train.flow_from_directory(DIR_PATH,target_size
        =(300,300),batch_size=32,
        class_mode='categorical',
        subset='training')
```

```

26 test_generator=test.flow_from_directory(DIR_PATH,target_size
    =(300,300),batch_size=32,
27                                     class_mode='categorical',
                                     subset='validation')
28 labels = (train_generator.class_indices)
29 labels = dict((v,k) for k,v in labels.items())
30
31 print(labels)
32
33 model=Sequential()
34
35 model.add(Conv2D(32,(3,3), padding='same',input_shape=(300,300,3)
    ,activation='relu'))
36 model.add(MaxPooling2D(pool_size=2))
37 model.add(Conv2D(64,(3,3), padding='same',activation='relu'))
38 model.add(MaxPooling2D(pool_size=2))
39 model.add(Conv2D(32,(3,3), padding='same',activation='relu'))
40 model.add(MaxPooling2D(pool_size=2))
41 model.add(Flatten())
42 model.add(Dense(64,activation='relu'))
43 model.add(Dense(6,activation='softmax'))
44
45 filepath="trained_model.h5"
46 checkpoint1 = ModelCheckpoint(filepath, monitor='val_acc',
    verbose=1, save_best_only=True, mode='max')
47 callbacks_list = [checkpoint1]
48 model.summary()
49 model.compile(loss='categorical_crossentropy', optimizer='adam',
    metrics=['acc'])
50 model.fit(train_generator, epochs=150, steps_per_epoch=2276//32,
    validation_data=test_generator,
51             validation_steps=251//32,callbacks=callbacks_list
    )
52 model.save("model.h5")
53
54 if __name__ == '__main__':

```



```
55     os.environ['TF_FORCE_GPU_ALLOW_GROWTH'] = 'true'
56     load_data()
```