Министерство образования и науки Российской Федерации

Федеральное государственное автономное образовательное учреждение

высшего профессионального образования

«Национальный исследовательский Нижегородский государственный университет им. Н.И. Лобачевского»

Институт Информационных Технологий, Математики и Механики

Направление: Фундаментальная информатика и информационные технологии

Магистерская программа: Когнитивные системы

**ОТЧЕТ**

по лабораторной работе №5

Тема:

**«Применение переноса обучения для решения задачи, поставленной во второй лабораторной работе»**

Выполнили:

Краснов Александр (гр. 381806м4)

Разин Вячеслав (гр. 381806м4)

Ахмедов Саркар (гр. 381806м4)

Проверила:

доцент, к.т.н. Кустикова В.Д.

Нижний Новгород

2019

Оглавление

[Введение 4](#_Toc33131211)

[Постановка задачи 5](#_Toc33131212)

[Описание набора данных 6](#_Toc33131213)

[Метрика качества решения задачи 7](#_Toc33131214)

[Подготовка данных для нейронной сети 8](#_Toc33131215)

[Выбор натренированной модели 9](#_Toc33131216)

[Тестовые конфигурации сетей 10](#_Toc33131217)

[Результаты экспериментов 13](#_Toc33131221)

[Анализ результатов 14](#_Toc33131222)

# Введение

Пусть - множество изображений, – множество номеров (или наименований) классов. Существует неизвестная *целевая зависимость* – отображение , значения которой известны только на объектах конечной обучающей выборки . Требуется построить алгоритм , способный классифицировать произвольный объект .

В рамках данной лабораторной работы ставится [задача классификации немецких дорожных знаков](https://www.kaggle.com/meowmeowmeowmeowmeow/gtsrb-german-traffic-sign) (The German Traffic Sign Benchmark).

# Постановка задачи

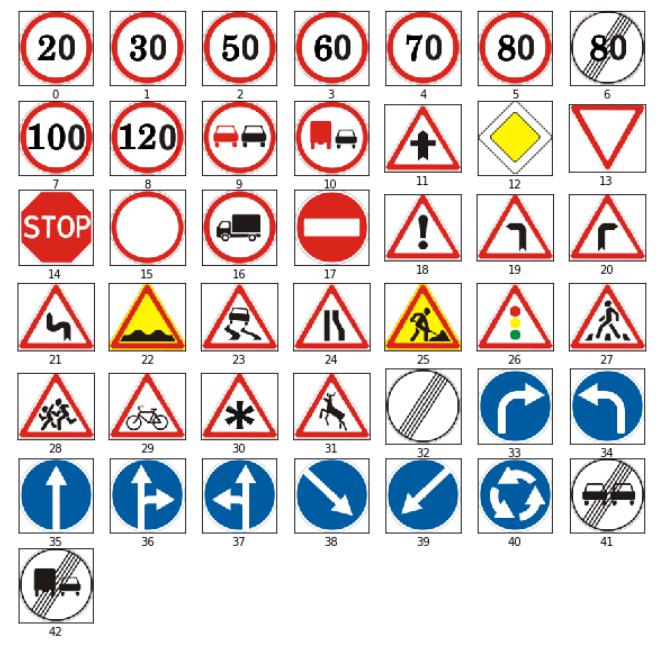
**Цели**

Цельнастоящей работы состоит в том, чтобы исследовать возможности переноса обучения для решения целевой задачи, выбранной изначально для выполнения практических работ.  
  
 **Задачи**

Выполнение практической работы предполагает решение следующих задач:

1. Поиск исходной задачи (близкой по смыслу к целевой задаче) и поиск натренированной модели для решения целевой задачи.
2. Выполнение трех типов экспериментов по переносу знаний (типы экспериментов описаны в лекции).

# Описание набора данных

Для лабораторных работ был выбран один из представленных на Kaggle набор данных <https://www.kaggle.com/meowmeowmeowmeowmeow/gtsrb-german-traffic-sign> (The German Traffic Sign Benchmark).

*Рис.1. Представители всех классов набора данных The German Traffic Sign Benchmark*

Данный набор содержит в общей сложности более 51840 изображений разных разрешений, 39209 из которых используются для тренировки нейронной сети, а 12631 - для тестирования. На каждом изображении содержится один дорожный знак. Набор данных содержит 43 класса изображений.

Для реализации различных архитектур нейронных сетей была выбрана открытая библиотека глубокого обучения Keras, написанная на языке программирования Python.

# Метрика качества решения задачи

Для оценки качества задачи классификации выбрана метрика "" (точность), показывающая отношение числа правильных предсказаний к числу предсказаний класса для всех изображений:  
где – количество изображений, – значения, выданные сетью при классификации изображений, – истинные значения классов элементов одежды с изображения.

# Подготовка данных для нейронной сети

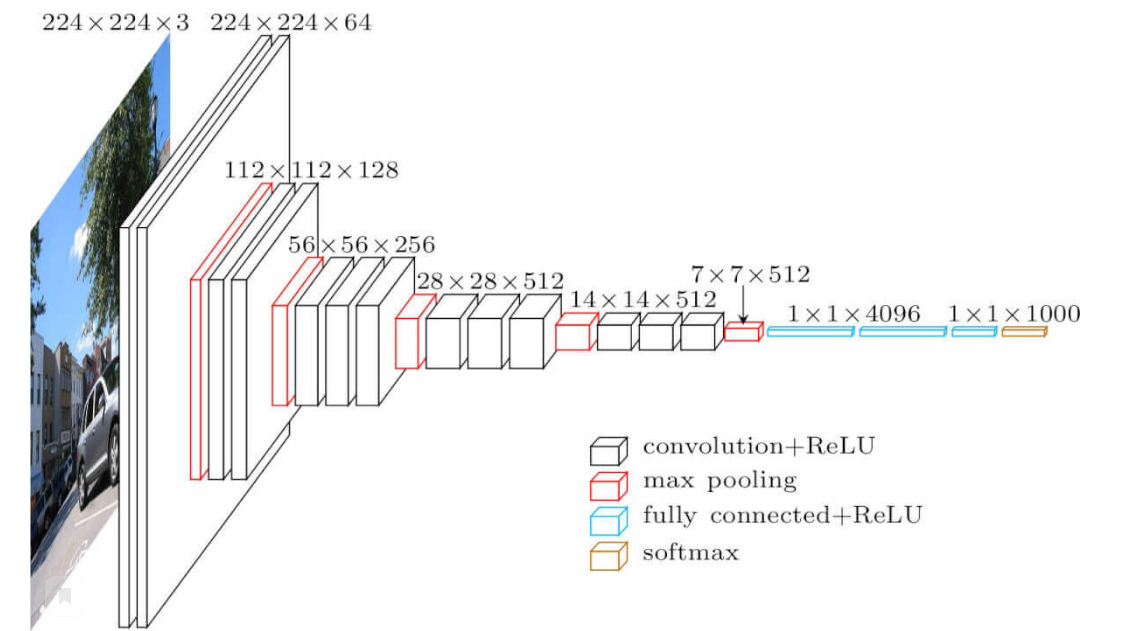
В силу большого количества входных изображений было решено сжать каждое из них до размера 50 x 50 пикселей. Сжатие происходит с сохранением соотношения сторон, уменьшенное изображение находится по центру, а недостающие пиксели имеют черный цвет. После этого преобразуем изображение (трёхмерный массив числе от 0 до 255 включительно) в один вектор и приводим каждый элемент вектора к значению с отрезка [0, 1].

Изначально каждое изображение обучающей выборки находится в пронумерованной папке, номер которой соответствует конкретному классу изображений от 0 до 42. Используя метод one-hot encoding, представляем каждый класс как вектор состоящий из 43 чисел. В итоге мы имеем два двумерных массива: массив изображений, готовых к классификации, и массив меток, отвечающий за то, какому классу данные изображения принадлежат.

# Выбор натренированной модели

В качестве натренированной модели для решения целевой задачи была выбрана нейронная сеть VGG-16.

VGG16 — модель сверточной нейронной сети, предложенная K. Simonyan и A. Zisserman из Оксфордского университета в статье “Very Deep Convolutional Networks for Large-Scale Image Recognition”. Модель достигает точности 92.7% — топ-5, при тестировании на ImageNet в задаче распознавания объектов на изображении. Этот датасет состоит из более чем 15 миллионов изображений, принадлежащих к 1000 классам.

VGG16 — одна из самых знаменитых моделей, отправленных на соревнование ILSVRC-2014. Она является улучшенной версией AlexNet, в которой заменены большие фильтры (размера 11 и 5 в первом и втором сверточном слое, соответственно) на несколько фильтров размера 3х3, следующих один за другим. Сеть VGG16 обучалась на протяжении нескольких недель при использовании видеокарт NVIDIA TITAN BLACK.

Архитектура сети VGG-16 представлена ниже:

*Рис.2. Архитектура VGG-16*

# Тестовые конфигурации сетей

Общая конфигурация сети:

|  |
| --- |
| Input (50 \* 50 RGB image) |
| Conv (3,3), 64, relu |
| Conv (3,3), 64, relu |
| **Maxpooling(2,2), 2** |
| Conv (3,3), 128, relu |
| Conv (3,3), 128, relu |
| **Maxpooling(2,2), 2** |
| Conv (3,3), 256, relu |
| Conv (3,3), 256, relu |
| Conv (1,1), 256, relu |
| **Maxpooling(2,2), 2** |
| Conv (3,3), 512, relu |
| Conv (3,3), 512, relu |
| Conv (1,1), 512, relu |
| **Maxpooling(2,2), 2** |
| Conv (3,3), 512, relu |
| Conv (3,3), 512, relu |
| Conv (1,1), 512, relu |
| **Maxpooling(2,2), 2** |
|  |
| FC – 1024, sigmoid |
| FC – 43, softmax |

**Эксперимент 1**

Использование структуры глубокой модели, построенной для решения исходной «Задачи А», с целью обучения аналогичной модели для решения «Задачи В»

* Предполагается, что модель, построенная для решения исходной задачи, обучается на данных, подготовленных для решения целевой задачи
* При этом веса модели инициализируются случайным образом
* Эксперимент реализует перенос знаний для родственных доменов

Т. е. в этом эксперименте используем только структуру модели. Чтобы применить данную модель VGG16 к нашей задаче, заменим полносвязный классификатор и заменим на последнем слое количество выходов с 1000 на 43.

**Эксперимент 2**

Использование модели, построенной для решения исходной «Задачи А», в качестве фиксированного метода извлечения признаков при построении модели, решающей «Задачу В»

* Идея данного подхода состоит в том, чтобы удалить из глубокой модели классификатор (последние полностью связанные слои) и рассматривать начальную часть сети как метод выделения признаков
* Взамен старого классификатора можно поместить новый классификатор (например, другой набор полностью связанных слоев или машину опорных векторов) и обучить его на признаках, построенных с использованием начальной части сети
* Эксперимент реализует перенос признакового описания

Во втором эксперименте возьмем сверточную основу VGG16, обученную на наборе ImageNet.

Пропустим наш набор данных через предварительно обученную сверточную основу VGG16, таким образом выделив признаки.

Далее возьмем полносвязный классификатор и обучим его на полученных признаках.

**Эксперимент 3**

Тонкая настройка параметров модели, построенной для решения исходной «Задачи А», с целью решения «Задачи В»

* Последние слои глубокой модели, соответствующие классификатору, который решает «Задачу А», заменяются новым классификатором (например, набором полностью связанных слоев с другим количеством выходов)
* Полученная модель обучается как единая система
* Эксперимент реализует перенос обучения на основе экземпляров

Последние слои глубокой модели VGG16, соответствующие классификатору, который решает ImageNet, заменим новым полносвязным классификатором с количеством выходов 43. И обучим модель как единую систему.

# Результаты экспериментов

Все вычислительные эксперименты выполнялись с помощью Google Colaboratory режиме GPU.

|  |  |  |  |
| --- | --- | --- | --- |
| Модель | 1  Использование только структуры глубокой модели | 2  Извлечение признаков и полносвязный классификатор | 3  Тонкая настройка |
| Батч | 256 | 256 | 256 |
| Количество эпох | 15 | 15 | 15 |
| Общее время (сек) | 480 | 479 | 480 |
| Точность (Accuracy) на тренировочном наборе, % | 99.83 | 99.66 | 98.64 |
| Ошибка на тренировочном наборе | 0.0098 | 0.0138 | 0.524 |
| Точность (Accuracy) на тестовом наборе, % | **97.50** | 96.77 | 94.84 |
| Ошибка на тестовом наборе | 0.1473 | 0.2040 | 0.2375 |

# Анализ результатов

В ходе выполнения лабораторной работы была получена модель 1, которая позволяет решать выбранную практическую задачу с достаточно высокими показателями качества, более высокими, чем на ранее рассмотренных моделях. В то же время можно сделать вывод, что использование переноса обучения с исходной задачи малоэффективно. С учетом затрат на обучение моделей получен небольшой прирост в точности.