

СТАТЬИ

УДК 378.016:004.8

DOI 10.17513/snt.40252

**ОРГАНИЗАЦИЯ ПРОЕКТНОЙ ДЕЯТЕЛЬНОСТИ
ПРИ ИЗУЧЕНИИ ТЕМЫ РАСПОЗНАВАНИЯ ИЗОБРАЖЕНИЙ
С ИСПОЛЬЗОВАНИЕМ НЕЙРОННЫХ СЕТЕЙ****Вильданов А.Н.***ФГБОУ ВО «Уфимский университет науки и технологий», Нефтекамский филиал,
Нефтекамск, e-mail: alvild@mail.ru*

В статье рассматривается методика проектного обучения студентов основам работы с технологиями компьютерного зрения. Целью исследования является формирование навыков создания нейронных сетей для решения задач классификации изображений с использованием библиотек Python, таких как TensorFlow, Keras, Pandas, NumPy и Matplotlib. В ходе работы последовательно рассматриваются три разные модели нейронных сетей: полносвязная сеть с двумя слоями, полносвязная сеть с тремя слоями и сверточная нейронная сеть. Проводится их сравнение по качеству распознавания изображений. Используются индивидуальные датасеты, сгенерированные с помощью программы «Дата Генератор», что обеспечивает разнообразие и репрезентативность обучающих выборок. В ходе выполнения проекта студенты разрабатывают и обучают нейронные сети, анализируя их производительность на валидационных данных, а также на самостоятельно подготовленных изображениях. С помощью сравнительного анализа обучающиеся убеждаются, что сверточная нейронная сеть демонстрирует наилучшие результаты по качеству распознавания, что подчеркивает ее преимущество в задачах компьютерного зрения. Проектная деятельность позволяет студентам глубже понять принципы работы нейронных сетей и способствует развитию практических навыков работы с современными библиотеками компьютерного зрения, что повышает качество подготовки специалистов в сфере современных информационных технологий.

Ключевые слова: студент, вуз, проектная деятельность, искусственный интеллект, компьютерное зрение, нейронная сеть, датасет, обучающая выборка, TensorFlow

**ORGANIZATION OF PROJECT ACTIVITIES
IN STUDYING THE TOPIC OF IMAGE RECOGNITION
USING NEURAL NETWORKS****Vildanov A.N.***Ufa University of Science and Technology, Neftekamsk branch,
Neftekamsk, e-mail: alvild@mail.ru*

The article discusses a methodology for project-based teaching students the basics of working with computer vision technologies. The aim of the study is to develop skills in creating neural networks to solve image classification problems using Python libraries such as TensorFlow, Keras, Pandas, NumPy and Matplotlib. In the course of the work, three different models of neural networks are successively considered: a fully connected network with two layers, a fully connected network with three layers, and a convolutional neural network. They are compared in terms of image recognition quality. Individual datasets generated using the Data Generator program are used, which ensures the diversity and representativeness of training samples. During the project, students develop and train neural networks, analyzing their performance on validation data, as well as on independently prepared images. With the help of comparative analysis, students are convinced that the convolutional neural network demonstrates the best results in recognition quality, which emphasizes its advantage in computer vision tasks. Project activities allow students to better understand the principles of neural networks and contribute to the development of practical skills in working with modern computer vision libraries, which improves the quality of training specialists in the field of modern information technology.

Keywords: student, university, project activities, artificial intelligence, computer vision, neural network, dataset, training sample, TensorFlow

Введение

В настоящее время искусственный интеллект является одной из наиболее перспективных и быстро развивающихся цифровых технологий, оказывающих значительное влияние на развитие общества и многие аспекты человеческой деятельности. Сегодня задача развития искусственного интеллекта стала важнейшей частью национальной стратегии Российской Федерации. Успехи технологий искусственного интеллекта

во многом определяются универсальностью подходов к совершенно различным задачам. Благодаря универсальности искусственного интеллекта можно успешно применять эти технологии в медицине, промышленности, машиностроении, сельском хозяйстве, образовании, геологии и т.д.

С развитием образовательных программ, которые стали включать в себя дисциплины, связанные с технологиями искусственного интеллекта, стала актуальной

проблема разработки задач на данную тематику, которые помогли бы студентам лучше понять теоретические аспекты технологий ИИ и приобрести навыки решения задач в данной сфере. Необходимы наглядные примеры и практические задачи, которые сделают процесс обучения более увлекательным, демонстрируя реальное применение технологий ИИ. Задачи должны способствовать развитию навыков программирования и аналитического мышления, в итоге это поможет готовить специалистов, способных работать с современными технологиями и решать сложные проблемы с использованием технологий ИИ.

Цель исследования заключается в разработке подхода к формированию у студентов навыков решения задач компьютерного зрения. В статье представлен алгоритм обучения в форме проектной деятельности, которая позволяет приобрести навыки работы с нейронными сетями на языке Python и освоить разработку алгоритмов решения задач компьютерного зрения с использованием библиотеки TensorFlow. Такой проектный подход способствует развитию самостоятельности студентов при решении

задач, организует регулярный обмен информацией между студентами и преподавателем, дает им представление о работе над реальными проектами, а также позволяет разработать прикладное программное решение по распознаванию изображений, которое обучающиеся смогут добавить в свое портфолио [1].

Материал и методы исследования

При проведении занятий в учебной группе желательно обеспечить разнообразие задач для большей активности каждого учащегося и выполнение им самостоятельной работы [2]. Поэтому сначала, на первом этапе, все студенты решают общий пример распознавания изображений. Далее, на втором этапе, каждый студент работает со своим вариантом классификации изображений. Каждый учащийся получает свой вариант датасета и проводит исследование самостоятельно. Это стало возможным благодаря генератору датасетов «Дата Генератор». Индивидуальный проект учащийся может выполнять в рамках курсового проекта, лабораторной или самостоятельной работы, и т.д.

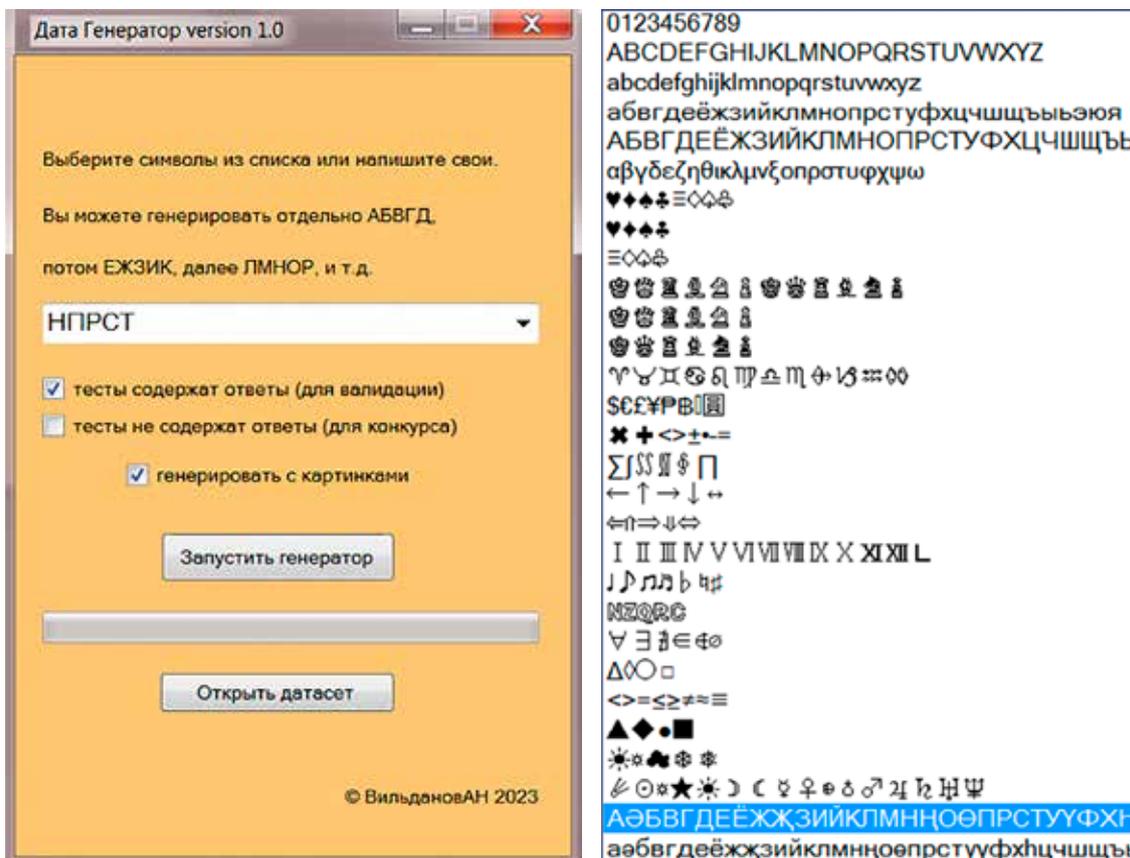


Рис. 1. Интерфейс программы «Дата Генератор»

Таблица 1

Этапы проектно-исследовательской деятельности

Этап № 1	<ul style="list-style-type: none"> • Студенты строят модель полносвязной двухслойной нейронной сети для решения задачи распознавания изображений согласных букв «Н», «П», «Р», «С», «Т» и производят ее обучение и валидацию. • Каждый студент самостоятельно готовит десяток собственных изображений букв «Н», «П», «Р», «С», «Т» и проверяет, как обученная сеть распознает их. • Далее обучающиеся строят модель полносвязной трехслойной нейронной сети и также производят ее обучение и валидацию. • Студенты снова проверяют, как новая обученная сеть распознает их собственные изображения. • Наконец, студенты строят модель сверточной нейронной сети и повторяют предыдущие шаги
Этап № 2	<ul style="list-style-type: none"> • Каждый студент работает с индивидуальным вариантом датасета для классификации изображений. Он самостоятельно продельвает все шаги предыдущего этапа, пытаясь построить наилучшую нейронную сеть. • По итогам исследования учащийся проводит сравнительный анализ, заполняя таблицу с результатами распознавания изображений разными моделями нейронных сетей

В проектной работе студенты строят и обучают нейронные сети разной сложности, начиная с полносвязных и заканчивая сверточными. Создание моделей нейронных сетей, их обучение и тестирование проводятся в облачной среде Google Colab с использованием библиотеки TensorFlow. На основе полученных результатов студенты анализируют качество распознавания изображений различными архитектурами нейронных сетей, выявляя преимущества сверточных сетей.

На завершающем этапе проекта каждый студент проводит самостоятельное исследование с использованием индивидуального варианта датасета. Формирование обучающей выборки (или датасета) – это важный шаг в процессе создания и обучения моделей машинного обучения. Датасет содержит данные, которые используются для того, чтобы «научить» модель выполнять конкретную задачу (например, классифицировать изображения, предсказывать значение стоимости или других числовых величин и т.д.).

При этом важно обеспечить репрезентативность обучающей выборки [3]. Для формирования заданий в рамках проектной деятельности используется программа «Дата Генератор», которая способна создавать датасеты по разнообразным тематикам для задач компьютерного зрения [4]. Интерфейс программы достаточно простой (рис. 1).

Для учебных задач достаточно генерировать 4–8 классов изображений. Тогда датасеты будут не слишком объемными, и процесс обучения нейронной сети будет проходить достаточно быстро.

В рассматриваемом ниже примере (рис. 1) генерируются датасеты для изо-

бражений пяти прописных согласных букв [«Н», «П», «Р», «С», «Т»].

Имеется также возможность выбора готовых вариантов. С помощью этой программы можно подготовить датасеты для задачи распознавания изображений цифр, русских, английских или греческих букв, музыкальных знаков, астрономических обозначений, карточных мастей, обозначений валют [4] и т.д.

Сгенерированный набор данных разбит на обучающую и тестовую выборку. Это позволяет проводить валидацию точности модели по различным метрикам [5].

План проектно-исследовательской деятельности обучающихся в области технологии компьютерного зрения

Предлагается следующий алгоритм выполнения обучающимися проектно-исследовательской деятельности в области технологий компьютерного зрения (табл. 1).

Как видно из таблицы 1, проект выполняется в два этапа. На первом этапе все студенты решают одинаковую задачу.

Набравшись опыта, каждый студент работает со своим вариантом датасета для классификации изображений. Такой подход позволяет сформировать у студентов навыки построения нейронной сети и облегчает достижение ими понимания сути этих сетей [6].

Результаты исследования и их обсуждение

В результате исследования подготовлен программный алгоритм выполнения проекта. Проект можно выполнять с помощью дистрибутива Anaconda или в Google Colab. Второй способ удобнее тем, что про-

грамма сохраняется в Google-аккаунте учащегося, и он сможет продолжить работу дома или же на следующем занятии. В данном проекте используется Google Colab, ссылка на файл проекта будет дана ниже.

Первым шагом в Google Colab следует создать новый «блокнот» (файл приложения) Jupyter Notebook. Работа начинается с подключения библиотек и их классов, загрузки и просмотра датасетов. Данный проект выполнен с использованием библиотек TensorFlow, Keras, Pandas, NumPy и Matplotlib. Рассмотрим подробнее их роль и взаимодействие.

В задачах искусственного интеллекта ключевую роль играет библиотека Pandas, поскольку работа с данными начинается с их загрузки и предобработки. В Pandas уже проделаны необходимые оптимизации, которые значительно ускоряют обработку даже очень больших объемов данных.

Табличные структуры данных, которые предоставляет Pandas, затем конвертируются в массивы NumPy для более эффективной и быстрой обработки.

После этого строится модель с использованием библиотек TensorFlow и Keras. TensorFlow – это библиотека для машинного обучения, разработанная Google, а Keras – это высокоуровневая библиотека для создания и обучения нейронных сетей, которая упрощает работу с TensorFlow. Взаимодействие происходит таким образом, что Keras позволяет пользователям легко строить нейронные сети через интуитивно понятный API, а TensorFlow выполняет всю низкоуровневую работу по вычислениям и оптимизации, обеспечивая высокую производительность.

TensorFlow и Keras облегчают быстрое прототипирование и экспериментирование с архитектурами нейронных сетей, позволяя разработчикам создавать и обучать модели с минимальной сложностью, тем самым служа ценным инструментом как для новичков, так и для опытных практиков в области глубокого обучения [7]. Они предоставляют разработчику удобный программный интерфейс для создания, обучения и предсказания нейронной сети.

Визуализация результатов выполняется с помощью библиотеки Matplotlib, что позволяет наглядно представить структуру данных и работу модели.

Итак, для выполнения проекта следует сначала загрузить необходимые библиотеки для работы с нейронными сетями, обработкой данных и их визуализацией: Keras, Pandas и NumPy для обработки данных, и Matplotlib для построения графиков. Команда `%matplotlib inline` позволяет отображать графики прямо в ноутбуке Jupyter или Google Colab (рис. 2).

Датасеты следует загрузить на Google-диск в отдельную папку. Когда датасет уже находится на Google-диске, его можно легко подключить к приложению Colab и использовать как источник данных для обучения моделей, анализа данных и других вычислительных задач (рис. 2).

Каждая строка в файле для обучения `train.csv` содержит информацию об одном изображении: первый столбец – это метка класса, остальные столбцы – пиксельные значения изображения. В нашем примере рассматриваются следующие пять классов русских заглавных согласных букв: `classes = [«Н», «П», «Р», «С», «Т»]`.

```
#загружаем нужные библиотеки
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import Dense, Flatten
from tensorflow.keras import utils
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
%matplotlib inline

Загружаем данные

[ ] from google.colab import drive
drive.mount('/content/drive/')
DIR = "2024/CV/dataset"
PATH = '/content/drive/My Drive/' + DIR + '/'
```

Рис. 2. Загрузка необходимых библиотек и подключение к Google-диск



Рис. 3. Датасет согласных букв

```
model = Sequential()
model.add(Dense(900, input_dim=400, activation="relu"))
model.add(Dense( 5, activation="softmax"))
```

Рис. 4. Модель полносвязной двухслойной сети

```
Epoch 1/16
58/58 [=====] - 2s 16ms/step - accuracy: 0.9513 - val_loss: 0.5525 - val_acc: 0.9877
Epoch 2/16
58/58 [=====] - 1s 12ms/step - accuracy: 0.9950 - val_loss: 0.6181 - val_acc: 0.9890
Epoch 3/16
58/58 [=====] - 1s 12ms/step - accuracy: 0.9952 - val_loss: 0.0689 - val_acc: 0.9939
Epoch 4/16
58/58 [=====] - 1s 13ms/step - accuracy: 0.9984 - val_loss: 0.0632 - val_acc: 0.9975
Epoch 5/16
58/58 [=====] - 1s 13ms/step - accuracy: 0.9952 - val_loss: 0.1927 - val_acc: 0.9963
...
Epoch 15/16
58/58 [=====] - 1s 12ms/step - accuracy: 1.0000 - val_loss: 0.1279 - val_acc: 0.9963
Epoch 16/16
58/58 [=====] - 1s 12ms/step - accuracy: 1.0000 - val_loss: 0.1279 - val_acc: 0.9963
```

Рис. 5. Ход обучения нейронной сети

После загрузки датасета можно для наглядности его визуализировать, чтобы посмотреть на него и убедиться, что мы работаем с нужными данными (рис. 3):

Далее обязательно проводится предварительная обработка данных изображений, включая нормализацию, стандартизацию и другие операции, которые помогут повысить качество обучения нейронной сети [8].

После подготовки данных создаются модели нейронных сетей и проводится их обучение. В проекте исследуются различные их архитектуры, такие как полносвязные нейронные сети с разным количеством слоев, и сверточные (CNN), которые лучше всего подходят для задач распознавания изображений. Обучение модели проводит-

ся на наборе данных train.csv с последующей проверкой на валидационном наборе для оценки ее производительности [9].

На первом шаге рассматривается полносвязная сеть из двух нейронных слоев. Модель такой нейронной сети строится с помощью класса `Sequential` (рис. 4).

Размер входных данных составляет 400 (так как у нас изображения размером 20 на 20 пикселей), функция активации на входном слое – `relu`. На первом, входном слое располагаются 900 нейронов. На выходном слое у нас должно быть 5 нейронов, так как это количество должно соответствовать количеству классов [9] (у нас ровно пять согласных букв: это «Н», «П», «Р», «С», «Т»).

Обучение созданной модели производится с помощью ее метода `fit` (рис. 5).

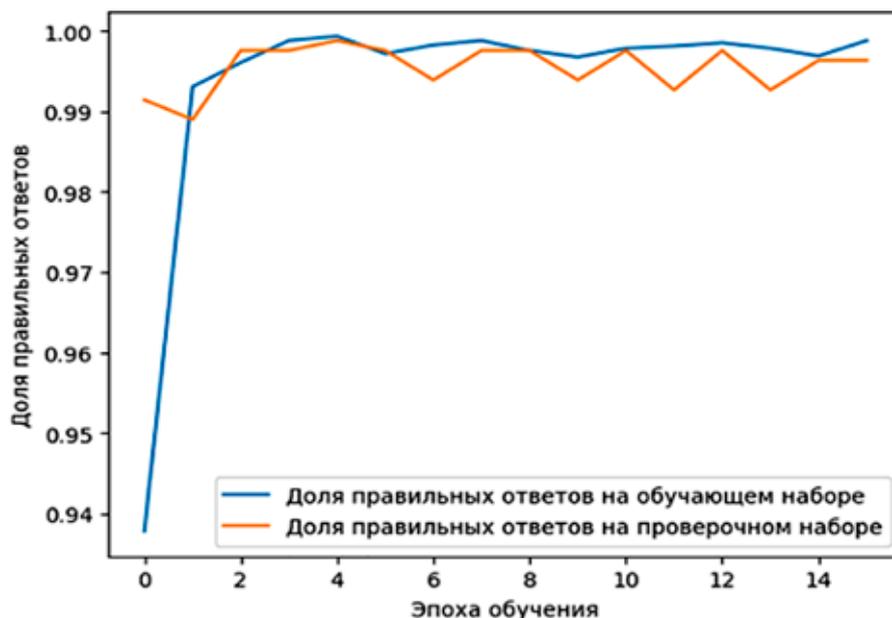


Рис. 6. График обучения нейронной сети



Рис. 7. Собственные тестовые изображения

Также можно вывести график, наглядно демонстрирующий процесс обучения (рис. 6). Обычно такие графики показывают две линии: одну для тренировочных данных (train) и другую для валидационных данных (validation). Если точность на тренировочных данных растет, но на валидационных данных перестает увеличиваться или даже начинает снижаться, это указывает на переобучение.

Таким образом, графики помогают наглядно понять, не начинает ли модель переобучаться, и сделать, например, такой вывод, что дальнейшее обучение бесполезно и имеет смысл перезапустить обучение с меньшим количеством эпох.

Теперь следует проверить, как нейронная сеть делает предсказание на валидаци-

онных и тестовых данных. Валидационные данные (датасет `validate.csv`) уже содержат правильные ответы. Они нужны для того, чтобы в автоматическом режиме оценить качество работы нейронной сети.

Для валидации модели нейронной сети применяется метод `evaluate`. Модель делает предсказания на валидационном наборе данных и преобразует их в числовые метки классов. Далее визуализируются первые 50 изображений из валидационного набора, где для каждого отображается предсказанный класс. Изображения выводятся в черно-белом формате, с подписями, указывающими результат распознавания, и при этом можно наглядно посмотреть, как работает нейронная сеть на валидационных данных.



Рис. 8. Тестирование работы двухслойной нейронной сети на собственных изображениях

```

from tensorflow.keras.layers import Conv2D, MaxPooling2D

model = Sequential()
model.add(Conv2D(32, (3, 3), input_shape=(img_width, img_height, 1),
                activation='relu'))
model.add(MaxPooling2D(pool_size=(2,2)))

model.add(Conv2D(64, (3, 3), activation='relu'))
model.add(MaxPooling2D(pool_size=(2, 2)))

model.add(Flatten())
model.add(Dense(200, activation='relu'))
model.add(Dense(5, activation='softmax'))

```

Рис. 9. Модель полносвязной трехслойной сети

```

model = Sequential()
model.add(Dense(900, input_dim=400, activation="relu"))
model.add(Dense(64, activation="relu"))
model.add(Dense(5, activation="softmax"))

```

Рис. 10. Модель сверточной сети

Далее учащийся самостоятельно подготавливает с помощью Paint собственные изображения для тестирования работы нейронной сети. Их можно закинуть в папку с датасетами.

Для последующего удобства подсчета вручную количества правильно распознанных изображений подготовлено их круглое количество – 10 изображений (рис. 7).

Далее следует преобразовать пиксели изображений в массивы `numpy` и провести предсказание. Для предсказания в модели применяется метод `predict`. Из рисунка ниже видно, что правильно предсказано 7 изображений из 10, то есть 70% изображений (рис. 8).

Далее можно немного усложнить нейронную сеть, сделав три слоя [9] (рис. 9).

Остальные шаги прорабатываются аналогично. Добавление слоя не дает особого

преимущества, а иногда может привести и к ухудшению качества предсказания.

В итоговой таблице 2 ниже показано, что трехслойная сеть правильно распознала только 6 изображений из 10, то есть 60% изображений. Можно сделать вывод, что полносвязные нейронные сети недостаточно качественно справляются с задачей распознавания изображений. Прогресс обучения двухслойной и трехслойной нейронной сети растет достаточно медленно из-за увеличения количества неизвестных параметров (весов) нейронной сети [9].

Более качественный результат можно получить, построив сверточную нейронную сеть. В структуре данного типа нейронных сетей используются слои свертки и подвыборки, чередующиеся между собой [10] (рис. 10).

Таблица 2

Качество распознавания изображений нейронными сетями

Структура НС	Количество весов для обучения	Качество на валидации	Качество распознавания собственных изображений
Полносвязная нейронная сеть из двух слоев	365405	97,75%	70%
Полносвязная нейронная сеть из трех слоев	372389	98,25%	60%
Сверточная нейронная сеть с двумя сверточными слоями	42629	99,65%	100%

Обучение такой сети проводится аналогично. На этот раз правильно предсказано 10 из 10, то есть 100% изображений.

После успешного выполнения и защиты работы каждый учащийся приступает к выполнению индивидуального проекта. Каждый учащийся получает свой вариант датасета и выполняет исследование самостоятельно, повторяя все шаги. Это поможет закрепить полученные навыки по распознаванию изображений и может выполняться в рамках курсового проекта, лабораторной или самостоятельной работы и т.д.

По результатам выполнения индивидуального проекта учащийся заполняет таблицу, которая содержит информацию о качестве распознавания изображений с использованием различных типов нейронных сетей, сравнивая их по следующим параметрам:

- 1) количество весов для обучения;
- 2) точность модели на валидационном наборе данных;
- 3) качество распознавания собственных изображений – точность распознавания изображений, которые были подготовлены обучающимися для тестирования непосредственно в ходе работы над проектом.

Такая таблица демонстрирует обучающимся преимущества сверточной нейронной сети перед полносвязной (табл. 2).

Заключение

Процесс обучения нейронной сети является важным этапом в задачах машинного обучения, особенно в компьютерном зрении, и требует обеспечения достаточности, разнообразия и равномерности классов в датасете.

В ходе выполнения проектной работы студенты поэтапно строят и обучают нейронные сети разной сложности, начиная с полносвязных и заканчивая сверточными. Они анализируют качество распознавания изображений различными архитектурами нейронных сетей, выявляя преимущества сверточных сетей. В конце проекта каждый студент выполняет самостоятельное исследование с использованием собственного варианта датасета.

Такой подход к организации проектной деятельности стал возможным благодаря программе «Дата Генератор», с помощью которой можно создавать разнообразные наборы данных для задачи распознавания изображений.

Список литературы

1. Государкин Я.С., Глазунов С.А. Проектное обучение школьников технологиям компьютерного зрения // Современное образование: содержание, технологии, качество. 2021. Т. 1. С. 308-311.
2. Вильданов А.Н. Построение интерактивных тестовых заданий по высшей алгебре с помощью mathjax // Современные наукоемкие технологии. 2022. № 12-2. С. 301-305. DOI: 10.17513/snt.39476.
3. Галушка В.В., Фатхи В.А. Формирование обучающей выборки при использовании искусственных нейронных сетей в задачах поиска ошибок баз данных // Инженерный вестник Дона. 2013. № 2. URL: ivdon.ru/magazine/archive/n2y2013/1597 (дата обращения: 12.10.2024).
4. Вильданов А.Н. Генерация датасетов для учебных задач компьютерного зрения // Инженерный вестник Дона. 2023. № 4(100). С. 92-101.
5. Тетеревенков Д.Л., Корчагин С.А. Математические модели и комплекс программ для определения эмоций человека с использованием технологий компьютерного зрения // Современные наукоемкие технологии. 2024. № 1. С. 82-86. DOI: 10.17513/snt.39912.
6. Нажимова Н.А., Наумова Е.Г. Формирование навыков построения нейронных сетей в различных программных средах // Современные проблемы науки и образования. 2024. № 3. URL: <https://science-education.ru/ru/article/view?id=33522> (дата обращения: 08.10.2024). DOI: 10.17513/spno.33522.
7. Яцкевич Е.С., Кушнир Н.В., Тотухов К.Е., Власенко А.В., Оганян А.Р., Высоцкий В.А. Анализ влияния нейронных сетей на программирование на Python // Политематический сетевой электронный научный журнал Кубанского государственного аграрного университета. 2024. № 198. URL: <http://ej.kubagro.ru/2024/04/pdf/42.pdf> (дата обращения: 12.10.2024). DOI: 10.21515/1990-4665-198-042.
8. Цзян С., Шэнь Ц., Яо Я. Распознавание изображений из мультфильмов на основе технологии глубокого обучения tensorflow // Современная наука: актуальные проблемы теории и практики. Серия: Естественные и технические науки. 2024. № 1. С. 146-148. DOI: 10.37882/2223-2966.2024.01.38.
9. Бадрисламов Д.И. Разработка сверточной нейронной сети для распознавания изображений карточной масти // Современные научные исследования и инновации. 2024. № 6 (158). [Электронный ресурс]. URL: <https://web.snauka.ru/issues/2024/06/102200> (дата обращения: 10.10.2024).
10. Сабиров А.И., Минниханов Р.Н., Каташев А.С., Муштафин Р.И. Нейросетевая система распознавания дорожных знаков // Вестник Технологического университета. 2024. Т. 27. № 1. С. 87-91. DOI: 10.55421/1998-7072_2024_27_1_87.