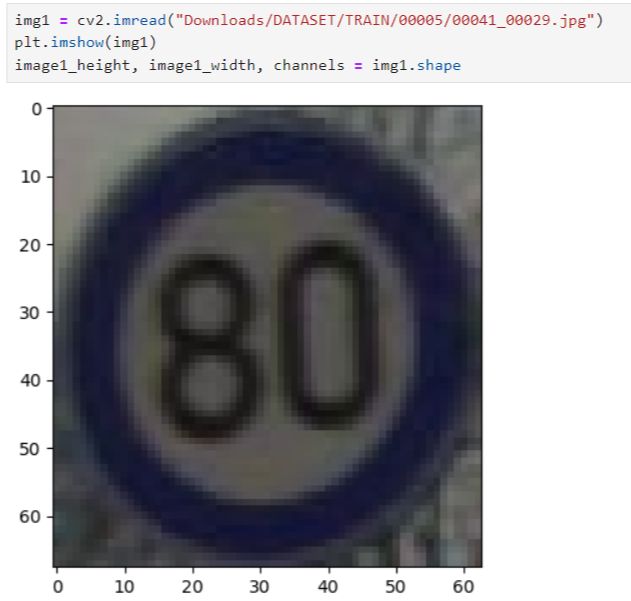
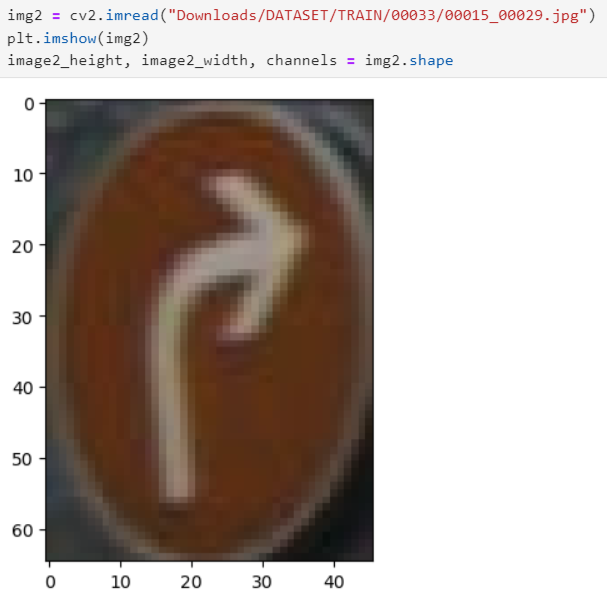
автор: Кремко Вера Валерьевна, 304 группа

Проект по распознаванию дорожных знаков

**Цель проекта** состоит в создании модели нейронной сети для распознавания дорожных знаков на изображениях.

Для этого использовался открытый набор данных с платформы kaggle, разделенный на тренировочное и тестовое множества, состоящий из 43 различных классов дорожных знаков, в каждом классе около 650 изображений, равномерное распределение.

<https://www.kaggle.com/datasets/shanmukh05/traffic-sign-cropped>

**Исходные данные** имеют формат jpg. Всего 26640 тренировочных и 12630 тестовых изображений различных исходных размеров

*примеры изображений*

**Предобработка данных** заключалась в использовании методов аугментации данных и подготовки генераторов данных для обучения и тестирования:

- масштабирование значения пикселей изображений в диапазон от 0 до 1

- случайное вращение изображения на угол до 20 градусов (для создания дополнительных вариантов изображений из существующих)

- случайное отражение изображения по горизонтали (расширение условий, на которых модель должна распознавать объекты)

- автоматическое разделение данных на тренировочный и валидационный наборы, где 20% данных используются для валидации

- перемешивание данных для каждой эпохи обучения

Размер изображений после изменения устанавливался 64х64

Предобработка модели помогает лучше обучаться на разнообразных условиях, что в конечном итоге приводит к более надежной и эффективной модели распознавания объектов на изображениях.

**Модель** представляет собой сверточную нейронную сеть (CNN) с несколькими сверточными слоями, слоями пулинга и полносвязными слоями для классификации изображений.

Слои свертки (‘Conv2D’):

1. 128 фильтров размером (3,3), функция активации ReLU, входной размер (64, 64, 3)
2. 64 фильтра размером (3,3), функция активации ReLU.
3. 128 фильтров размером (3,3), функция активации ReLU.
4. 256 фильтров размером (3,3), функция активации ReLU.

Пулинговые слои (‘MaxPooling2D’):

1. размер пула (2,2) с шагом (strides) 2.
2. размер пула (2,2) с шагом (strides) 2.
3. размер пула (2,2) с шагом (strides) 2.
4. размер пула (2,2) с шагом (strides) 2.

Flatten(): слой, преобразующий выходы последнего сверточного слоя в одномерный вектор перед подачей на полносвязные слои

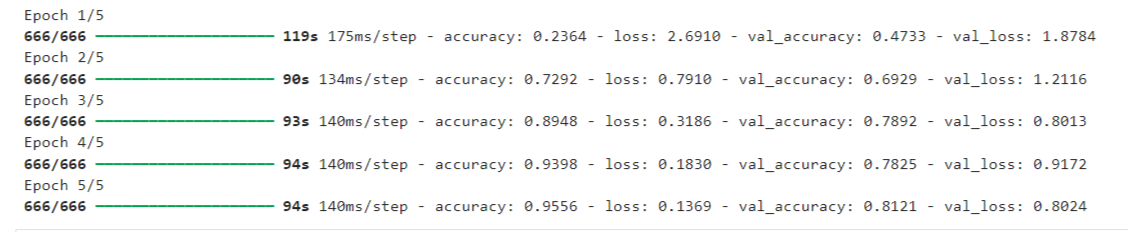
Полносвязные слои (‘Dense’):

1. 64 нейрона с функцией активации ReLU.
2. 128 нейронов с функцией активации ReLU.
3. Выходной слой: количество нейронов равно числу классов, в данном случае 43, функция активации - softmax для многоклассовой классификации.

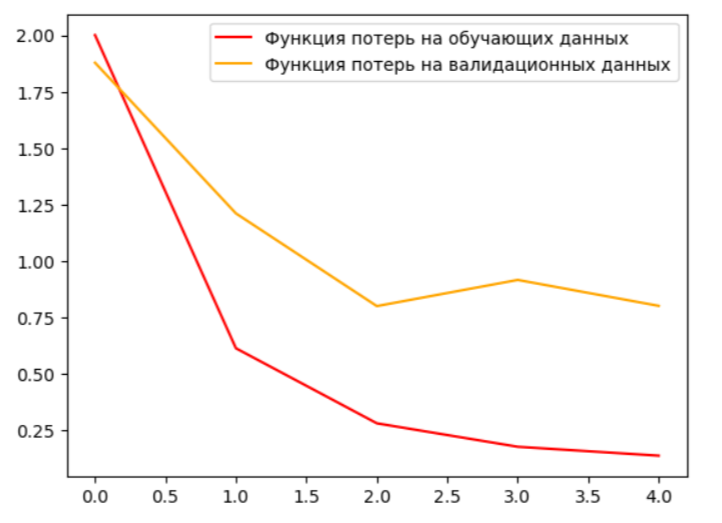
Размер пакета данных, передаваемых на каждой итерации обучения равен 32 (batch\_size=32). Используется оптимизатор Adam, который обновляет веса модели на основе градиентов функции потерь, в качестве функции потерь - категориальная кросс-энтропия, определяет разницу между предсказанными и истинными метками классов.

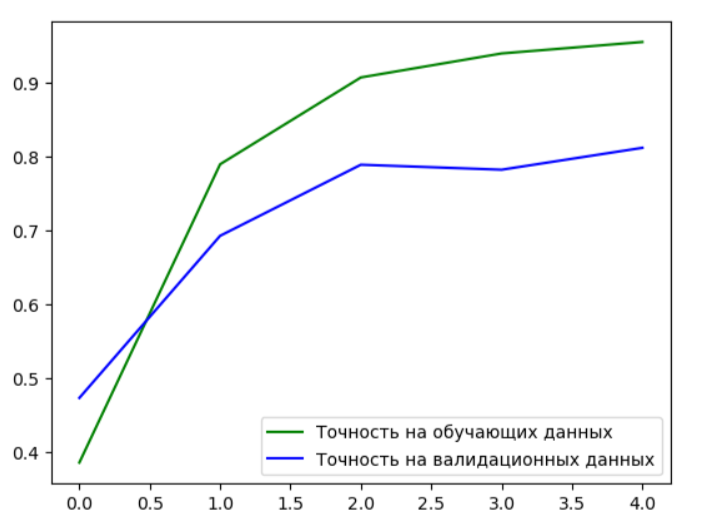
**В качестве метрики** оценки качества модели выбрана точность (‘accuracy’), которая измеряет долю правильно предсказанных классов.

Модель была обучена в течение 5 эпох. Каждая эпоха состояла из 666 шагов с временем обучения 90-120 секунд на эпоху. Первая эпоха показала низкую точность на уровне 23.64% на обучающем наборе и высокую функцию потерь (2.6910). На валидационном наборе точность была выше (47.33%). Следующие эпохи показали улучшение производительности модели. Точность на обучающем наборе увеличилась до 95.56% к концу 5-ой эпохи, а функция потерь снизилась до 0.1369.



*информация о каждой эпохе обучения*





*графики функции ошибки и метрики качества*

**Программа** написана на языке Python в программе Jupyter Notebook. Состоит из основных частей:

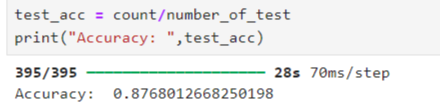
1. Подключение библиотек
2. Импортирование данных
3. Визуализация данных
4. Обработка данных
5. Создание модели
6. Обучение модели
7. Тестирование модели
8. Примеры распознавания некоторых элементов

Использованы **библиотеки:**

1. TensorFlow для создания, обучения и развертывания нейронных сетей
2. Matplotlib для визуализации данных: построения графиков функций потерь, метрик и других данных в процессе обучения нейронных сетей.
3. Pandas для загрузки, обработки и подготовки данных перед обучением моделей
4. NumPy для вычислений, предобработки данных, поддержки многомерных массивов (тензоров)
5. Keras для построения и обучения нейронных сетей, работающий поверх TensorFlow. Позволяет определять модели как последовательные структуры слоев
6. Os для взаимодействия с файловой системой компьютера и выполнения различных операций с файлами и директориями
7. Cv2 для загрузки, обработки, анализа и изменения изображений, включая операции с пикселями, фильтрацию изображений

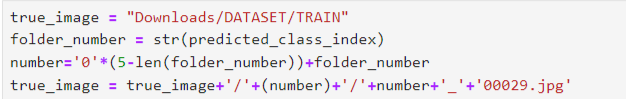
**Проблемы** возникли с оценкой точности на тестируемых изображениях, появлялась ошибка при вызове model.evaluate(), так как тестовый датасет отличался по структуре от тренировочного, в нем все элементы находились в одной папке, не было разделения по классам, из-за чего модель воспринимала все изображения как один вид дорожных знаков.

**Для решения** проблемы функция вычисления точности была написана самостоятельно, использовался файл test\_labels.csv. Он содержит в себе информацию о том, к какому классу относится каждая фотография из тестовых данных. Эти метки загружаются в виде массива, затем подсчитывается количество правильных предсказаний: сравниваются предсказанные моделью метки с истинными. Точность вычисляется как отношение числа правильных предсказаний к общему количеству объектов в тестовом наборе



Также возникла **проблема** с последней частью кода – «примеры использования некоторых элементов». В модель загружается определенная фотография дорожного знака, происходит предсказывание (model.predict()), после чего выдается класс дорожного знака. По одному номеру класса не всегда понятно, какой в итоге знак распознала модель. Данных о том, какой знак сопоставляется номеру класса датасет не содержит

**Решением** было выводить случайное изображение из папки того класса, который определила модель. Имя изображения имеет вид «ххххх», где с конца записывается его номер в папке класса, на остальных местах нули. Путь записан как true\_image



где predicted\_class\_index – предсказанный моделью индекс  
00029 - случайный номер картинки, выбранный заранее (учитывается, что 29ое изображение есть во всех классах)

В результате получаем такой вывод



Код

**Итог:** в рамках проекта по распознаванию дорожных знаков была поставлена задача распознавания на основе набора данных, содержащего изображения различных дорожных знаков. Для обучения и тестирования модели было использовано разделение на обучающее и валидационное множества, а также были предоставлены метки (классы) для изображений, использована сверточная нейронная сеть.

Проект демонстрирует потенциал нейронных сетей в распознавании дорожных знаков, дальнейшие улучшения могут привести к повышению точности и обобщающей способности модели.

Код приложен к отчету в формате .ipynb