

МОСКОВСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ  
ИМЕНИ М. В. ЛОМОНОСОВА

Отчёт по программе  
«Распознавание символов» Вариант №5

Антипова Алёна  
325 группа

Москва  
2024

## Постановка задачи

В данном проекте требуется реализовать простую сверточную нейронную сеть (CNN) для распознавания двух-трех символов.

Необходимо выполнить следующие подзадачи:

- создать датасет для символов, которые в дальнейшем будет распознавать программа;
- реализовать аугментацию данных датасета;
- построить архитектуру модели, при необходимости изучив дополнительные материалы;
- проанализировать результаты.

## Создание датасета

Для распознавания были выбраны буквы латинского алфавита K, Z и M, так как в них присутствуют схожие графические элементы. В директории **dataset** (для обучения и тестирования модели) находится по 10 изображений для каждой буквы. В директории **test** (для применения обученной модели) - по 3 изображения для каждой буквы. Каждое изображение является чёрно-белым, на белом фоне нарисованы чёрные буквы, и имеет размер 32 на 32 пикселя. Расширение файлов **.png**.

Изображения рисовались вручную с использованием ресурсов, предлагаемых онлайн-сервисом <https://www.piskelapp.com/>.

Путь до изображений для обучения имеет вид:

**dataset/[letter\_name]([letter\_number]).png.**

Например, **dataset/K (1).png**

Путь до изображений, с которыми работает обученная модель, имеет вид:

**test/test\_[test\_number].png.**

Например, **test/test\_1.png**

## Аугментация данных

К данным для обучения программно добавлялись аугментированные данные. Код аугментации и вывод полученных с помощью неё изображений, как и реализация всего изначального задания, можно увидеть в ноутбуке **notebook.ipynb**.

После загрузки исходных данных создается их дублированная версия в переменных **data\_augmented** и **labels\_augmented**. Это необходимо для применения аугментации к существующим данным без изменения оригинальных изображений.

Генератор аугментации данных создается с использованием класса **ImageDataGenerator** из библиотеки **Keras**. В данном коде применяются следующие методы аугментации:

- **rotation\_range**: случайное вращение изображения на угол до 20 градусов.
- **width\_shift\_range**: случайное смещение изображения по ширине до 20% от его размера.
- **height\_shift\_range**: случайное смещение изображения по высоте до 20% от его размера.
- **shear\_range**: случайное сдвигание изображения.
- **zoom\_range**: случайное увеличение или уменьшение изображения до 20%.
- **horizontal\_flip**: не применяется (изображения не переворачиваются).
- **fill\_mode**: режим заполнения пустых пикселей при трансформациях (используется ближайшее значение).

Для каждого исходного изображения генерируется пять аугментированных версий. Каждое изображение преобразуется в нужный формат (4D массив), чтобы соответствовать требованиям генератора. Сгенерированные изображения добавляются в список **augmented\_images**, а их метки — в список **augmented\_labels**.

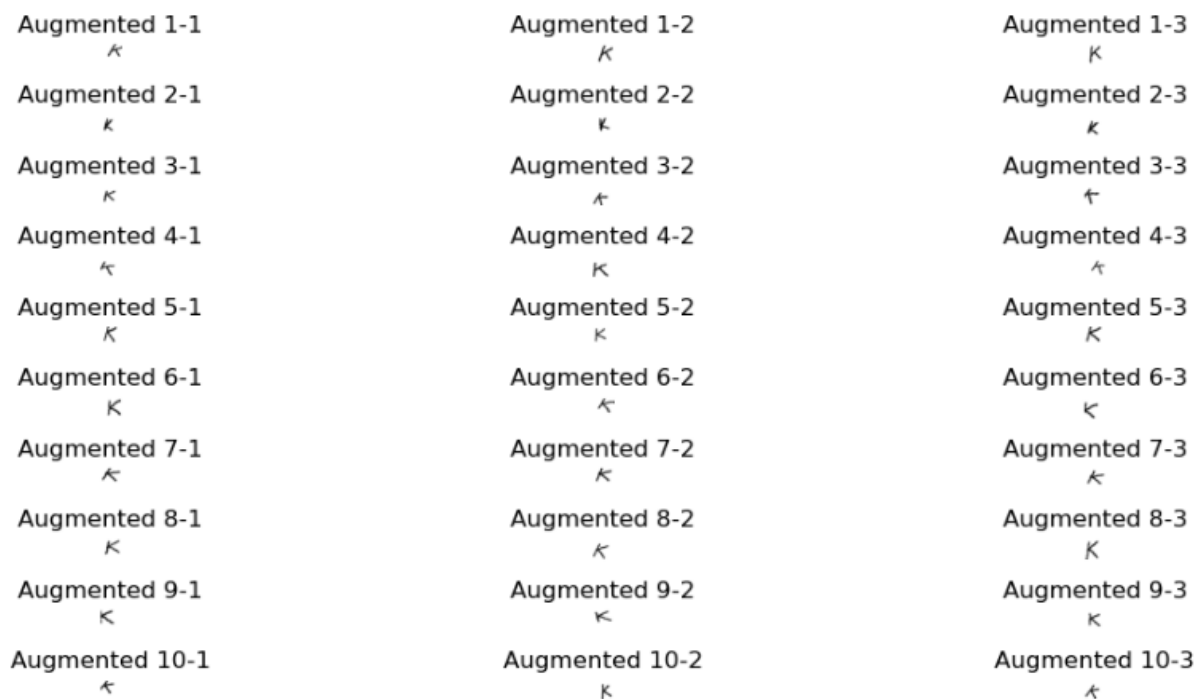


Рис. 1. Пример аугментированных данных.

## Описание программы

В качестве дополнительных материалов для написания программы был выбран следующий [источник](#).

В этом гайде подробно объясняется, как работать со свёрточными нейронными сетями (CNN). Этот вид нейронных сетей был выбран для написания программы, так как он был специально разработан для обработки изображений и видео и идеально подходит для задач классификации изображений, распознавания объектов и сегментации.

## Необходимые библиотеки

- **PIL** — для открытия и обработки изображений
- **numpy** — для хранения и обработки данных изображений
- **tensorflow** — для построения модели
- **matplotlib** — для отображения изображений
- **keras** — для генерации аугментированных изображений во время обучения.

## Выбранная архитектура

Архитектура модели сверточной нейронной сети (CNN) включает несколько ключевых компонентов. Входные данные преобразуются в формат с добавленной размерностью для черно-белых изображений. Модель строится с использованием последовательного подхода.

Первый сверточный слой с 32 фильтрами размером 3x3 извлекает признаки из изображений, за ним следуют слои подвыборки, которые уменьшают размерность данных. Далее добавляются дополнительные сверточные слои с увеличением числа фильтров (64 и 128), что позволяет извлекать более сложные признаки. После этого данные преобразуются в одномерный массив с помощью слоя Flatten. Полносвязный слой с 256 нейронами обрабатывает извлеченные признаки, а слой Dropout предотвращает переобучение. Завершает модель выходной слой с функцией активации softmax для многоклассовой классификации.

## Анализ результатов

На первоначальных данных сеть работала незначительно хуже. **Loss** на 20% тестовых данных без аугментации составляла 0.151862 против 0.030762 с аугментацией.

Методом перебора было получено достаточное количество эпох для обучения модели - 15.

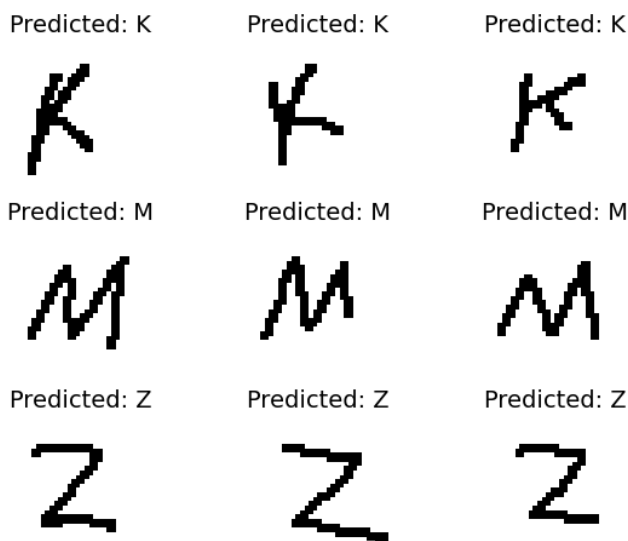


Рис. 2. Пример предсказания моделью символов.

## Выводы:

В ходе работы была построена и обучена сверточная нейронная сеть, решающая задачу распознавания изображений с символами К, Z и М. Применение аугментации значительно улучшило обобщающую способность модели. Итоговая сеть распознаёт все изображения в контрольной выборке корректно (9 из 9).

Программа является гибкой: при замене изображений и корректной подготовке нового датасета модель можно переобучить на другие символы. Данный проект может служить базой для более масштабных систем распознавания рукописного ввода.