

РАЗРАБОТКА ПРИЛОЖЕНИЯ ДЛЯ ПРЕОБРАЗОВАНИЯ ЯЗЫКА ЖЕСТОВ В ТЕКСТ

Нагорный В.О., студент гр. ПИБ-212, IV курс,
Базыль В.О., студент гр. ПИБ-212, IV курс
Научный руководитель: Глебова Е.А., ст. преп.
Кузбасский государственный технический университет
имени Т.Ф. Горбачева, г. Кемерово

Язык жестов - эффективное средство коммуникации для людей с нарушениями слуха и речи. С развитием технологий создание приложений, способных преобразовывать язык жестов в текст, становится все более актуальным. Такие приложения могут значительно улучшить взаимодействие между людьми, использующими жестовый язык, и теми, кто не владеет этой системой коммуникации.

Основной целью разработки приложения является создание инструмента, который сможет в реальном времени распознавать жесты и преобразовывать их в текст, обеспечивая при этом точность и скорость обработки информации. Для начала мы создали обучающий и тестовый набор данных. После идет создание модели для обучения. Здесь мы использовали сверточную нейронную сеть (CNN) [1] для построения этой модели, представленную на рисунке 1.

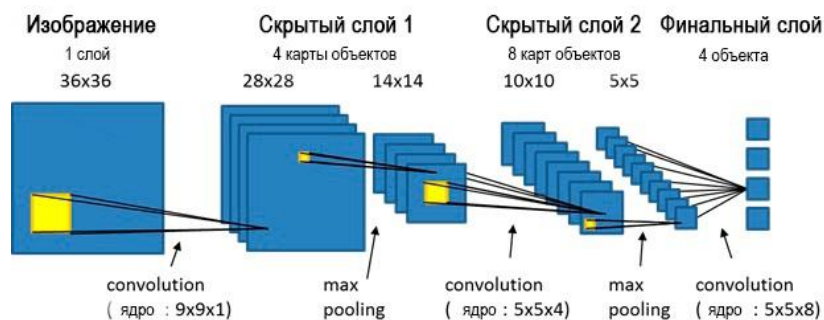


Рисунок 1 – Сверточная нейронная сеть

Краткое изложение модели следующее: Сверточная нейронная сеть (CNN). В отличие от обычных нейронных сетей, в слоях CNN нейроны расположены в трех измерениях: ширина, высота, глубина. Нейроны в слое будут связаны только с небольшой областью слоя (размером с окно) перед ним, а не со всеми нейронами полностью связанными. Более того, конечный выходной слой будет иметь размерность (количество классов), поскольку к концу архитектуры CNN мы сведем все изображение к одному вектору оценок классов. В нашем проекте используется компьютерное зрение [2], основным элементом которого является выделение пикселей из изображения, чтобы изучить объекты и, таким образом, понять, что они содержат. Ключевые аспекты, которые компьютерное зрение стремится распознать на изображениях: обнаружение

объектов, распознавание объектов, различные виды классификации, сегментация, то есть выделение пикселей, принадлежащих объекту. В проекте все это реализуется с помощью библиотеки OpenCV пример которой представлен на рисунке 2.

```
gray = cv2.cvtColor(cv2image, cv2.COLOR_BGR2GRAY)

blur = cv2.GaussianBlur(gray, ksize: (5, 5), sigmaX: 2)

th3 = cv2.adaptiveThreshold(blur, maxValue: 255, cv2.ADAPTIVE_THRESH_GAUSSIAN_C, cv2.THRESH_BINARY_INV, blockSize 11, C: 2)

ret, res = cv2.threshold(th3, thresh: 70, maxval: 255, cv2.THRESH_BINARY_INV + cv2.THRESH_OTSU)
```

Рисунок 2 – Код для обработки изображения

В коде преобразуется цветное изображение в оттенки серого, затем применяем Гауссово размытие к серому изображению для уменьшения шума. После применяется адаптивная пороговая бинаризация, чтобы выделить объекты на изображении. В результате получается двоичное изображение, где объекты на относительно светлом фоне становятся белыми. Таким образом, общий смысл данного кода заключается в том, чтобы взять изображение, преобразовать его в градации серого, размыть, а затем применить адаптивное пороговое преобразование для выделения объектов на изображении.

Часть кода отвечающая за распознавание жеста с помощью нейронной модели представлена на рисунке 3.

```
def predict(self, test_image):
    test_image = cv2.resize(test_image, dsize: (128, 128))
    result = self.loaded_model.predict(test_image.reshape(1, 128, 128, 1))
    prediction = {}
    prediction['blank'] = result[0][0]
    inde = 1
    for i in ascii_uppercase:
        prediction[i] = result[0][inde]
        inde += 1
    prediction = sorted(prediction.items(), key = operator.itemgetter(1), reverse = True)
    self.current_symbol = prediction[0][0]
```

Рисунок 3 – код для распознавания жеста

Для начала мы передаем обработанное изображение в модель для предсказания, на выходе мы записываем вероятности принадлежности к разным буквам в словарь (prediction), где ключ – это буква алфавита, а значение – вероятность. После мы сортируем словарь и выбираем значение с наибольшей вероятностью – это наш распознанный символ. Пример распознавания жеста на рисунке 4.



Рисунок 4 – Распознавание жеста

Во время тестирования мы обнаружили что приложение путается в распознавании некоторых похожих жестов. В будущем мы планируем реализовать дополнительные нейронные модели для тех групп жестов, в которых путается основная модель, чтобы уточнять какая именно буква правильная. Также добавить функцию для исправления или предсказания слова, что поможет предотвратить возможные опечатки или строить новые слова на основе уже введенного текста.

Список литературы:

1. Материалы сайта «Наглядно о том, как работает свёрточная нейронная сеть» [Электронный ресурс]. – Режим доступа: <https://habr.com/ru/companies/skillfactory/articles/565232/> (дата обращения 27.03.2025) - Режим доступа: свободный.
2. Шакирьянов, Э.Д. Компьютерное зрение на Python. Первые шаги/ Москва «Лаборатория знаний», 2021. – 163 с.