ОЛИМПИАДА ШКОЛЬНИКОВ «ШАГ В БУДУЩЕЕ»
ПО ПРОФИЛЮ «ИНЖЕНЕРНОЕ ДЕЛО»

38085

регистрационный номер

Компьютерные системы
название секции

Нейросеть для распознавания текста на изображении и его интерпретация для людей с ограниченными возможностями
название работы

Автор: ***Казаков Иван Сергеевич***
фамилия, имя, отчество
ГБОУ Школа №1155, 10
наименование учебного заведения, класс

Научный руководитель: ***Юсупова Кристина Олеговна***
фамилия, имя, отчество
ГБОУ Школа №1155
место работы
учитель информатики
звание, должность

_____
подпись научного руководителя

Москва – 2024

Нейросеть для распознавания текста на изображении и его интерпретация для людей с ограниченными возможностями

Аннотация

Целью работы является создание нейросети для распознавания текста на изображении для устройства, интерпретирующего текст людям с ограниченными возможностями.

Создание программы произведено в среде разработки Visual Studio 2022. Реализация алгоритмов выполнена на языке C++.

Созданная нейросеть анализирует полученное на вход изображение, выявляя на нем печатный текст, и транслирует его пользователю, который имеет ограничения по зрению, в доступном для него формате. Для качественной передачи информации с изображения в проекте было создано устройство, которое будет переводить распознанный нейросетью текст в Шрифт Брайля. Разработанная система на этапе тестирования подтвердила свою работоспособность. Для этого была создана обучающая выборка, прописан алгоритм определения отдельных букв, далее было произведено усложнение до определения букв в составе слов. После готовности алгоритма была разработана модель устройства в TinkerCAD и создан ее демонстрационный макет для тестирования функционала с помощью Arduino IDE.

СОДЕРЖАНИЕ

ВВЕДЕНИЕ

В данной работе представлен инструмент, с помощью которого незрячие и плохо видящие люди могут получать текстовую информацию с изображения. В работе представлена программа, выполняющая идентификацию букв в словах на изображении, с помощью нейросети распознавание каждой из этих букв и дальнейшая сборка букв в единый связный текст.

Данный алгоритм был заложен в основу устройства, который состоит из набора ячеек с выдвижными элементами, которые выдвигаются согласно шрифту Брайля после получения на вход сигнала о новой распознанной нейросетью буквой. Другими словами, происходит перенос печатного текста в автоматизированные ячейки Брайля. Плюсом такого подхода служит то, что данное устройство работает как электронная книга: достаточно загрузить туда новый отсканированный или напечатанный текст, и человек может приступить к чтению тактильного шрифта.

Нейросеть, составляющая основу программы, которая распознает текст по изображению, была написана, обучена и протестирована на языке C++ в среде разработки Visual Studio. С помощью программы TinkerCAD была выполнена схема соединений устройства, а с помощью среды разработки Arduino IDE была создания и загружена полная программа, обеспечивающая работоспособность устройства, на Arduino-совместимую плату.

Целью работы является создание нейросети для распознавания текста на изображении для устройства, интерпретирующего текст людям с ограниченными возможностями.

Для достижения поставленной цели были выполнены следующие задачи:

1. создание обучающей выборки для нейросети;

2. создание нейросети, ее обучение и тестирование;

3. разработка алгоритма по определению букв в словах для дальнейшего их определения нейросетью и перевода в печатный текст;

4. создание демонстрационной физической модели устройства, в основе которого лежит созданная программа по переводу бук на изображении в доступный пользователю вид;

5. тестирование модели, корректировка программы и элементов устройства.

ОСНОВНАЯ ЧАСТЬ

1 Этапы создания программы

Самым важным и сложным этапом данного проекта является написание собственной нейронной сети на языке C++, которая будет распознавать в тексте отдельные буквы и выводить им текстовый аналог для дальнейшей передачи на устройство.

На рисунке 1 представлена схема нейросети, полный цикл ее работы для распознавания одной буквы с изображения до получения результата.

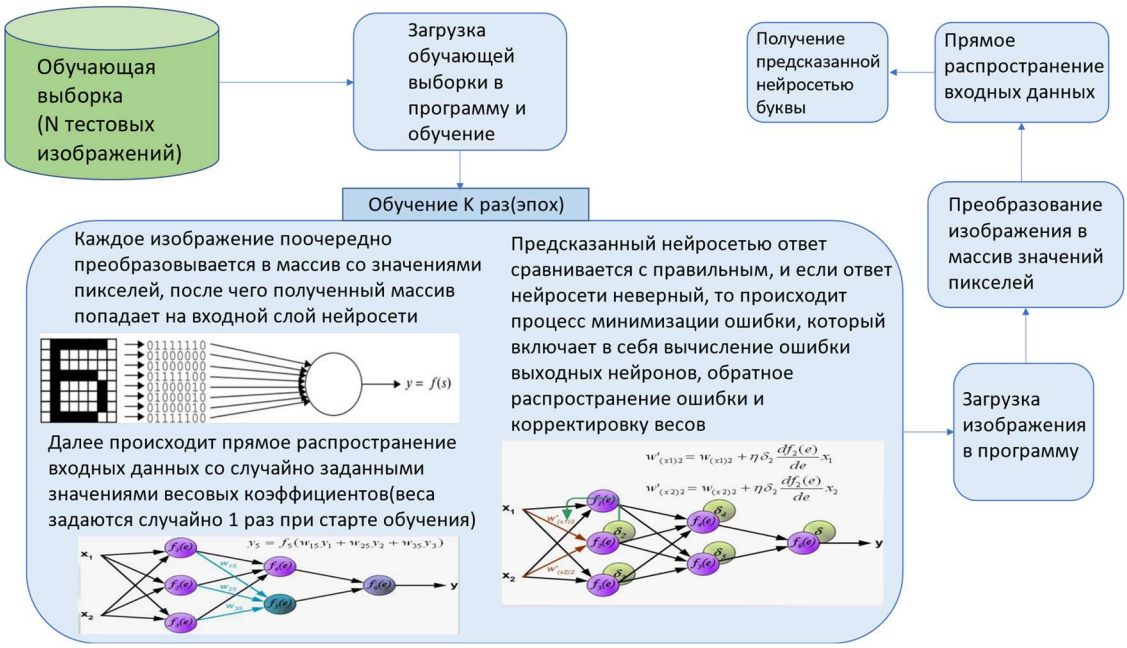


Рисунок 1 – Схема работы нейронной сети

Как видно по рисунку 1 самым первым шагом, который влияет на эффективность определения буквы, является этап обучения нейросети. Обучение происходит на большом массиве данных: в нейросеть подается много

изображений всех букв с разным начертанием и много правильных ответов к этим изображениям. С помощью заложенных в модель математических основ нейросеть учится выдавать правильные ответы и на другие изображения похожего вида [1, 2]. Для того, чтобы нейросеть с высокой точностью выдавала верные ответы на тестовые изображения, было произведено ее глубокое обучение: на большом количестве обучающих изображений разных букв в разных начертаниях, для гибкости инструмента относительно различных шрифтов (рис. 2).

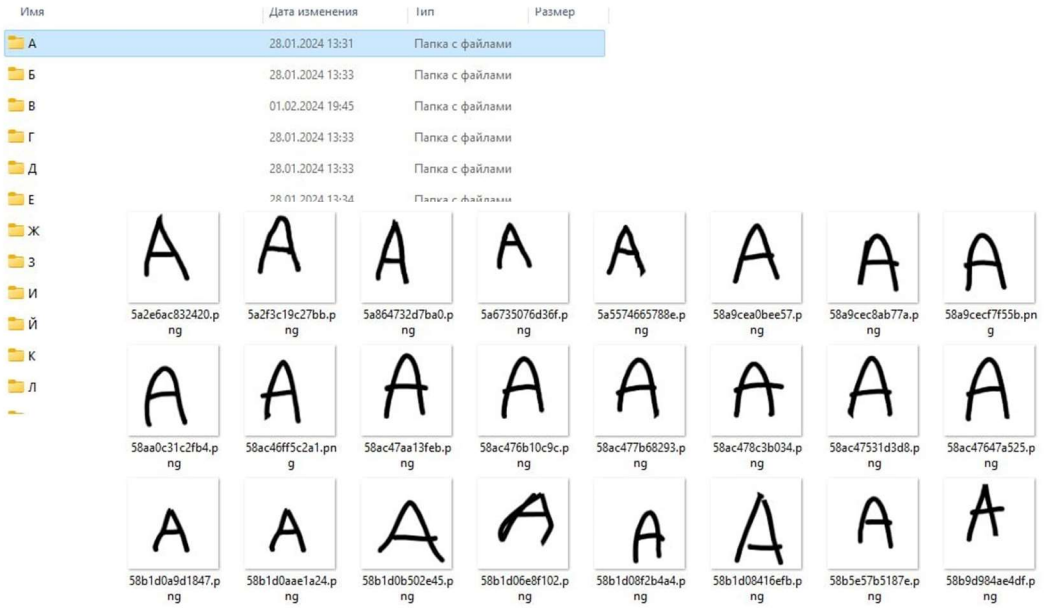


Рисунок 2 – Обучающая выборка

Стоит пояснить, на основе каких математических методов создана нейронная сеть в данной работе. Для корректировки весов используется метод градиентного спуска, чтобы уменьшить ошибки нейронов. Градиент – это вектор, который указывает в направлении наискорейшего возрастания функции. Так как для обучения нейросети нужно уменьшать значение ошибки нейронов, то используется антиградиент. Таким образом, цель всего процесса – получение наименьшего значения функции потерь (ошибки). Чтобы найти самое низкое значение этой функции, используется производная функции активации [3].

Приведем также описание кода программы, ключевых фрагментов, которые являются реализацией нейронной сети по шагам, представленным на схеме рисунка 2. На рисунке 3 описывается структура нейрона, которая включает в себя значение (value), ошибку (error) и функцию активации [1, 2].

```
struct neuron
{
    double error = 0;
    double value = 0;
    void act() {
        if (value < 0) value = value * 0.01;
        else if (value > 1) value =  1. + 0.01 * (value - 1.);
    }
};
```

Рисунок 3 – Структура нейрона

На рисунке 4 представлен класс для работы с нейросетью, а также функция для инициализации нейросети. Эта функция включает в себя инициализацию матрицы весов, матрицы нейронов, количество слоев нейронов и количество нейронов на каждом слое. Также в этой функции задаются случайные весовые коэффициенты.

```
class Network
{
public:
    double*** ves;
    neuron** neurons;
    int lays;
    int* neuonlay;
    int skolvo;
    double* error;
    void initicialization(int layses, int* nela) {
        srand(time(0));
        lays = layses;
        neuonlay = new int[layses];
        neurons = new neuron * [layses];
        ves = new double** [layses - 1];
        for (int lay = 0; lay < layses; lay++) {
            if (lay == layses - 1) {
                error = new double[nela[lay]];
                error[nela[lay]] = { 0 };
            }
            neuonlay[lay] = nela[lay];
            neurons[lay] = new neuron[nela[lay]];
            if (lay < layses - 1) {
                ves[lay] = new double* [nela[lay]];
                for (int ne = 0; ne < nela[lay]; ne++) {
                    ves[lay][ne] = new double[nela[lay + 1]];
                    for (int w = 0; w < nela[lay + 1]; w++) {
                        ves[lay][ne][w] = ((rand() % 100)) * 0.03 / (nela[lay] + 35);
                    }
                }
            }
        }
    }
```

Рисунок 4 – Класс для работы с нейросетью

```
double fact(double x) {
    if (x < 0) return x * 0.01;
    else if (x > 1) return 1. + 0.01 * (x - 1.);
}
double fder(double x) {
    if (x < 0 || x > 1)
        return 0.01;
    else
        return 1;
}
```

Рисунок 5 – Функция активации

Во фрагменте кода на рисунке 5 представлена функция активации, а также производная этой функции. В качестве функции активации используется модифицированная *Relu*(*x*). *Relu*(*x*) – это стандартная функция активации, которая в зависимости от значения нейрона *x* возвращает *max*(0, *x*). Но в рамках нашей задачи был необходим диапазон значений нейрона *x* от -1 до 1, поэтому к этой функции было добавлено условие, чтобы перевести все значения в нужный диапазон [1, 2].

```
void savevesa(string path) {
    ofstream vesa(path);
    for (int lay = 0; lay < lays - 1; lay++) {
        for (int ne = 0; ne < neuonlay[lay]; ne++) {
            for (int w = 0; w < neuonlay[lay + 1]; w++) {
                vesa << ves[lay][ne][w] << " ";
            }
            vesa << endl;
        }
    }
}
void loadvesa(string path) {
    ifstream f;
    f.open(path);
    for (int lay = 0; lay < lays - 1; lay++) {
        for (int ne = 0; ne < neuonlay[lay]; ne++) {
            for (int w = 0; w < neuonlay[lay + 1]; w++) {
                f >> ves[lay][ne][w];
            }
        }
    }
}
```

Рисунок 6 – Функции сохранения и загрузки весов нейросети

На рисунке 6 описаны две функции. Одна для сохранения весов после обучения нейросети в текстовый файл, другая для загрузки весов из файла в нейросеть.

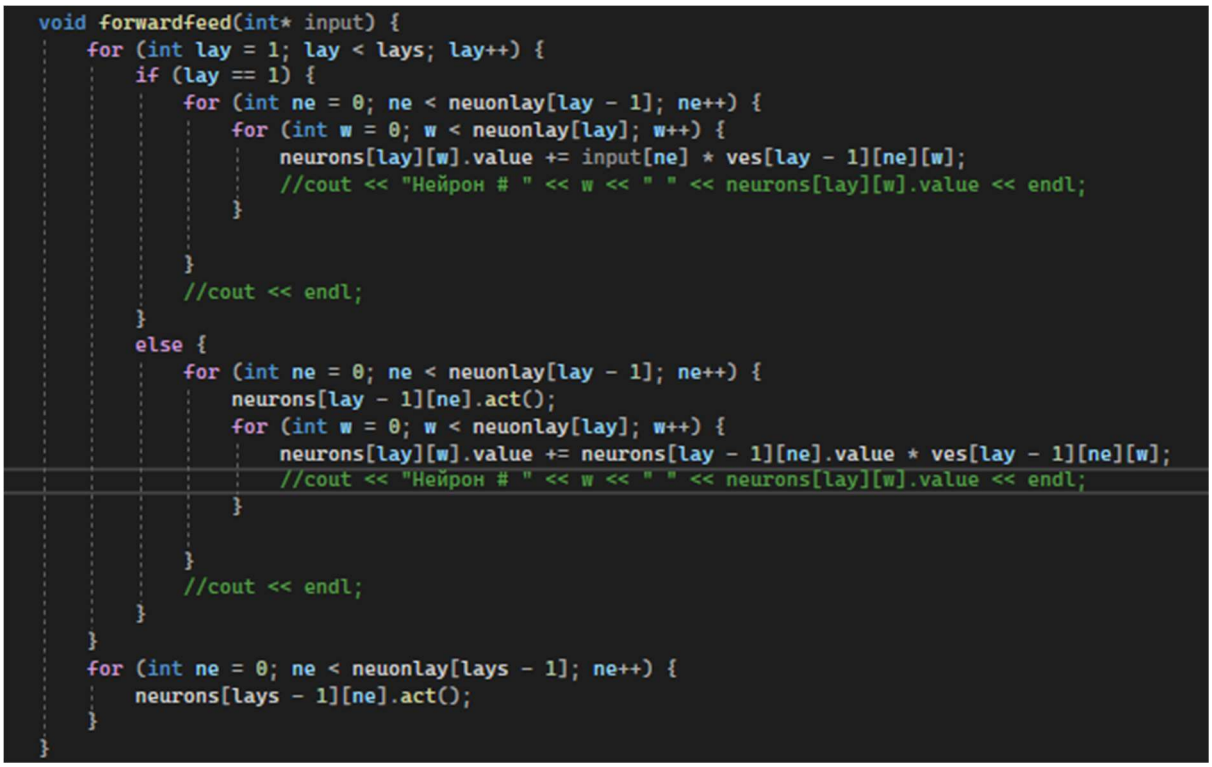


Рисунок 7 – Функция прямого распространения входных данных

На рисунке 7 представлена функция прямого распространения входных данных по всем слоям нейронной сети. Значение каждого нейрона (начиная со 2 слоя) – это сумма активированных (прошедших через функцию активации) значений нейронов на предыдущем слое, которые умножены на соответствующие им веса.

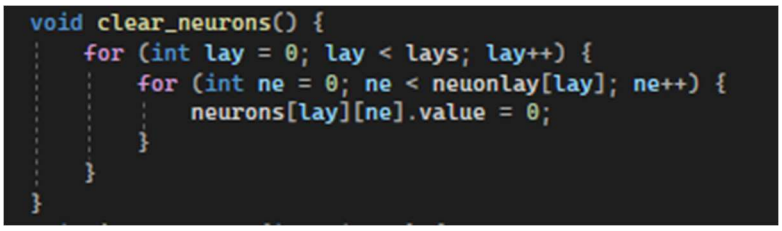
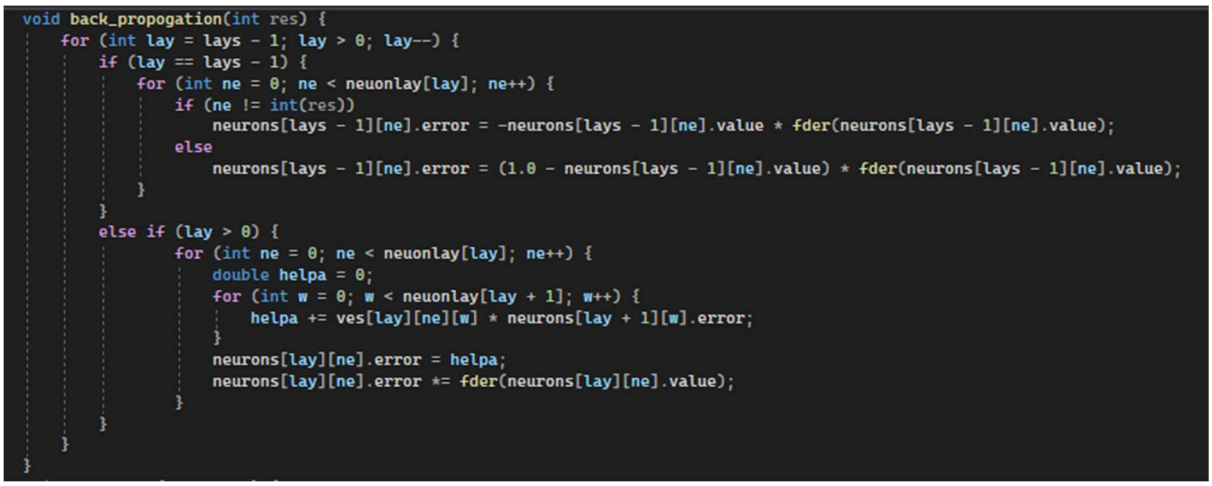


Рисунок 8 – Функция для обнуления значений всех нейронов

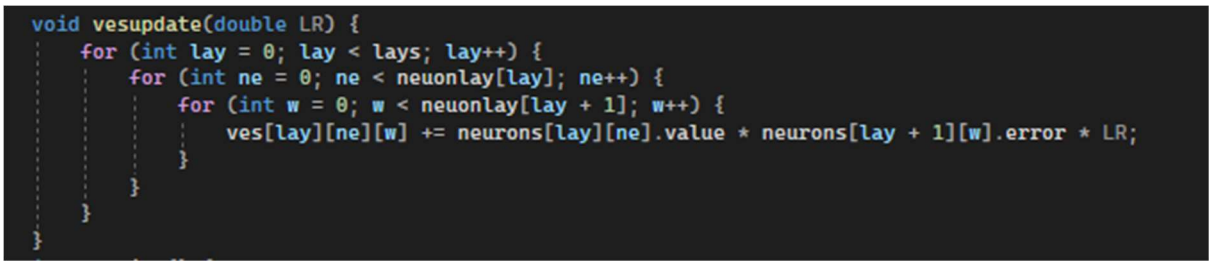
Во фрагменте кода на рисунке 8 представлена функция для обнуления значений всех нейронов. Она применяется после каждого примера в обучении.



```
void back_propogation(int res) {
    for (int lay = lays - 1; lay > 0; lay--) {
        if (lay == lays - 1) {
            for (int ne = 0; ne < neuonlay[lay]; ne++) {
                if (ne != int(res))
                    neurons[lays - 1][ne].error = -neurons[lays - 1][ne].value * fder(neurons[lays - 1][ne].value);
                else
                    neurons[lays - 1][ne].error = (1.0 - neurons[lays - 1][ne].value) * fder(neurons[lays - 1][ne].value);
            }
        }
        else if (lay > 0) {
                for (int ne = 0; ne < neuonlay[lay]; ne++) {
                    double helpa = 0;
                    for (int w = 0; w < neuonlay[lay + 1]; w++) {
                        helpa += ves[lay][ne][w] * neurons[lay + 1][w].error;
                    }
                    neurons[lay][ne].error = helpa;
                    neurons[lay][ne].error *= fder(neurons[lay][ne].value);
                }
        }
    }
}
```

Рисунок 9 – Функция обратного распространения ошибки

На рисунке 9 код функции обратного распространения ошибки. Сначала считается величина ошибки выходных нейронов – это разница между правильным значением из обучающей выборки и полученным нейросетью значением. Затем, используя ошибки выходного слоя нейронов и производную функции активации, считаются ошибки для всех остальных нейронов.



```
void vesupdate(double LR) {
    for (int lay = 0; lay < lays; lay++) {
        for (int ne = 0; ne < neuonlay[lay]; ne++) {
            for (int w = 0; w < neuonlay[lay + 1]; w++) {
                ves[lay][ne][w] += neurons[lay][ne].value * neurons[lay + 1][w].error * LR;
            }
        }
    }
}
```

Рисунок 10 – Функция для обновления весов

На рисунке 10 описана функция для обновления весов таким образом, чтобы уменьшить ошибки нейронов. Также в вычислениях используется скорость обучения (LR, learning rate), которая задается опытным путем в процессе обучения нейросети.

```
    while (ra / examples < 0.99) {
        ra = 0;
        auto t1 = chrono::steady_clock::now();
        for (long long i = 0; i < skolvo; i++) {
            net.inputneurons(inputdata[i]);
            net.forwardfeed(inputdata[i]);
            int pred = net.predict();
            //  cout << i << endl;
            int rightanswer = rightanswers[i];
            // cout << pred << " " << rightanswer << endl;
            if (pred != rightanswer) {
                net.back_propogation(rightanswer);
                net.vesupdate(LR * exp(-epoch / 20.));
            }
            else if (pred == rightanswer) {
                ra++;
            }
            net.clear_neurons();
        }
        auto t2 = chrono::steady_clock::now();
        time = t2 - t1;
        if (ra > maxra)maxra = ra;
       //cout << "ra: " << ra / examples * 100 << "\t" << "maxra: " << maxra / examples * 100 << "\t"
        epoch++;
        if (epoch == 80) {
            net.savevesa(vesa);
            net.forwardfeed(test1);
            cout << net.predict() << endl;
            break;
        }
    }
    auto end = chrono::steady_clock::now();
    time = end - begin;
    cout << "TIME: " << time.count() / 60. << " min" << endl;
}
```

Рисунок 11 – Обучение нейросети

Во фрагменте кода на рисунке 11 описано обучение нейросети. На каждый пример из обучающей выборки применяется функция прямого распространения (forwardfeed), затем функция обратного распространения ошибки и функция обновления весов, после чего значения нейронов обнуляются. Обучение идет 80 итераций (эпох), на каждой итерации все примеры из обучающей выборки проходят вышеописанные действия.

```
for (int y = 0; y < img.rows; y++) {
    for (int x = 0; x < img.cols; x++) {
        int b = img.at<Vec3b>(y, x)[0];
        int g = img.at<Vec3b>(y, x)[1];
        int r = img.at<Vec3b>(y, x)[2];
        if (b >= 0 && b <= 250) {
            test.push_back(1);
            cout << 1 << " ,";
        }
        else if (b == 255) {
            test.push_back(0);
            cout << 0 << ", ";
        }
        else {
            test.push_back(0);
            cout << 0 << " ,";
        }
    }
    cout << endl;
}
int* test1 = new int[784];
for (int i = 0; i < test.size(); i++) {
    test1[i] = test[i];
}
```

Рисунок 12 – Преобразование изображения с буквой в массив

На рисунке 12 показано, как изображение с буквой преобразуется в массив со значениями пикселей. После чего этот массив подается в функцию прямого распространения. Буквой будет являться номер нейрона с наибольшим значением (А – 1, Б – 2 и так далее).

2 Устройство, обеспечивающее возможность чтения тактильного шрифта

В проекте также разработано устройство, в основу которого входит созданная и описанная ранее нейронная сеть. Работа программы в составе устройства будет иметь структурную схему, изображенную на рисунке 13.

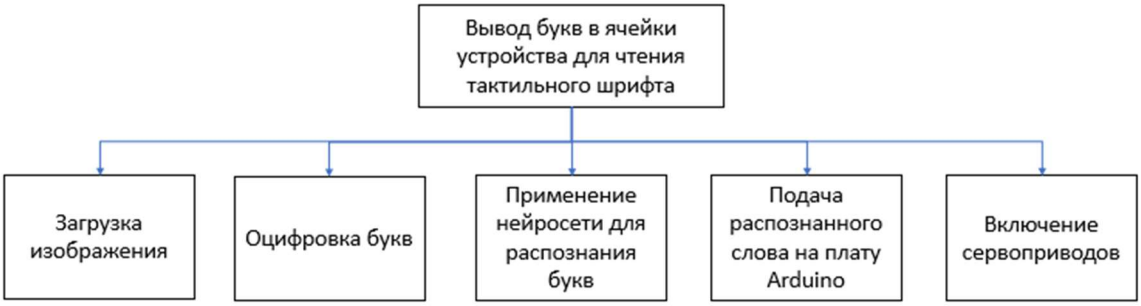


Рисунок 13 – Структурная схема программы

Приведем схему подключения для созданной модели устройства (рис. 14).

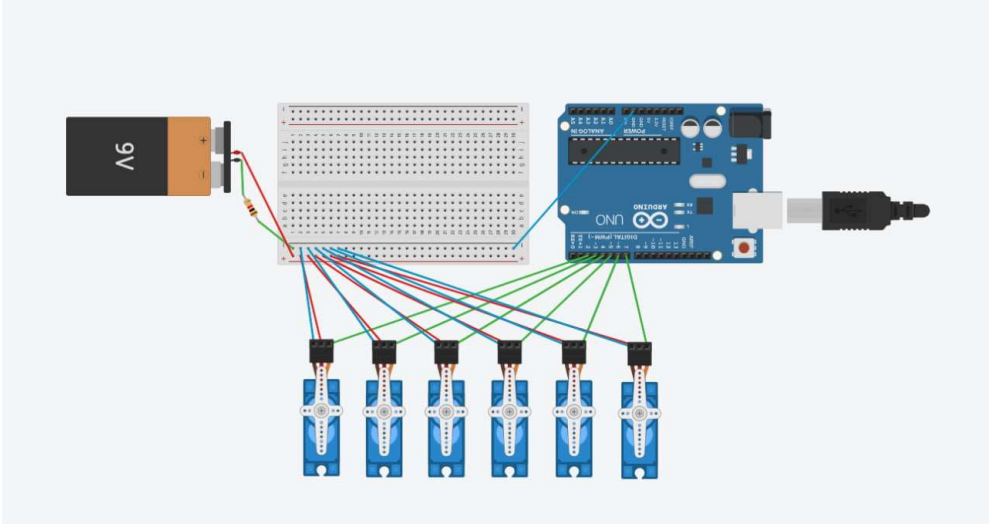


Рисунок 14 – Схема устройства

Схема выполнена в программе TinkerCAD, которая является эмулятором Arduino и позволяет моделировать электрические цепи. Перечислим основные элементы, составляющие реальную работающую сборку модели устройства согласно схеме: преобразователь напряжения, плата Arduino nano, сервопривод с углом поворота 180º (6 штук), соединительные провода, макетная плата. Внутреннюю структуру сборки из всех перечисленных инструментов покажем на рисунке 15

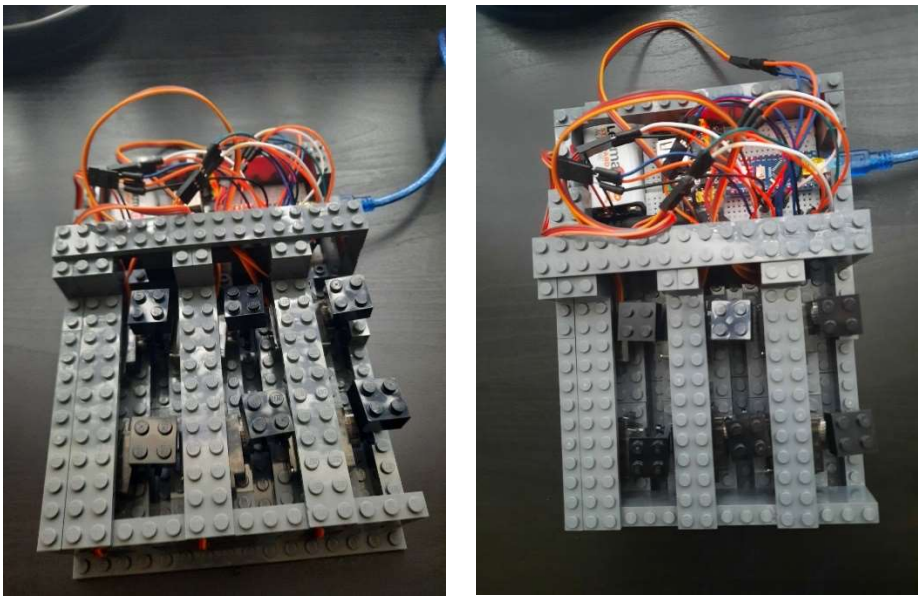


Рисунок 15 – Сборка демонстрационной модели

Опишем принцип работы устройства. На Arduino через COM-порт подается распознанное нейросетью слово, затем в зависимости от букв срабатывает нужный ключ и поворачиваются нужные сервоприводы для выдвижения деталей.

```
var = Serial.read();
switch(var){
  case 'A'{
    servo1.write(90);
  }
  case'B'{
    servo1.write(90);
    servo2.write(90);
  }
```

Рисунок 16 – Команды, приводящие в движение нужные сервоприводы

На рисунке 16 представлен фрагмент кода в Arduino IDE, который обеспечивает выдвижение подвижных элементов с помощью сервоприводов, соответствующих разным буквам (рис. 17).

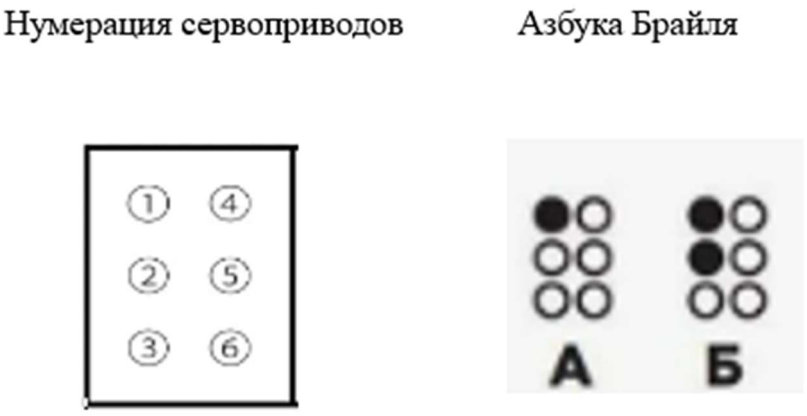


Рисунок 17 – Соответствие сервоприводов элементам азбуки Брайля

Азбука Брайля представляет собой таблицу уникальных ячеек для каждой буквы алфавита. В каждой ячейке присутствует шесть рельефных точек, определенный набор из которых имеет выпуклую структуру для того, чтобы незрячий человек мог тактильно идентифицировать букву. Например, если в слове встретится буква «Б», то придут в движение сервоприводы под номером 1 и 2, выдвинутся соответствующие им детали, что соответствуют букве «Б» в азбуке Брайля.

Завершим описание разработанной системы по интерпретации печатного текста со снимков в удобный незрячим людям формат. На рисунке 18 представлена функциональная схема модели со всеми шагами обработки изображения, распознавания отдельных букв в словах, поочередное выдвижение подвижных элементов устройства под соответствующие буквы.

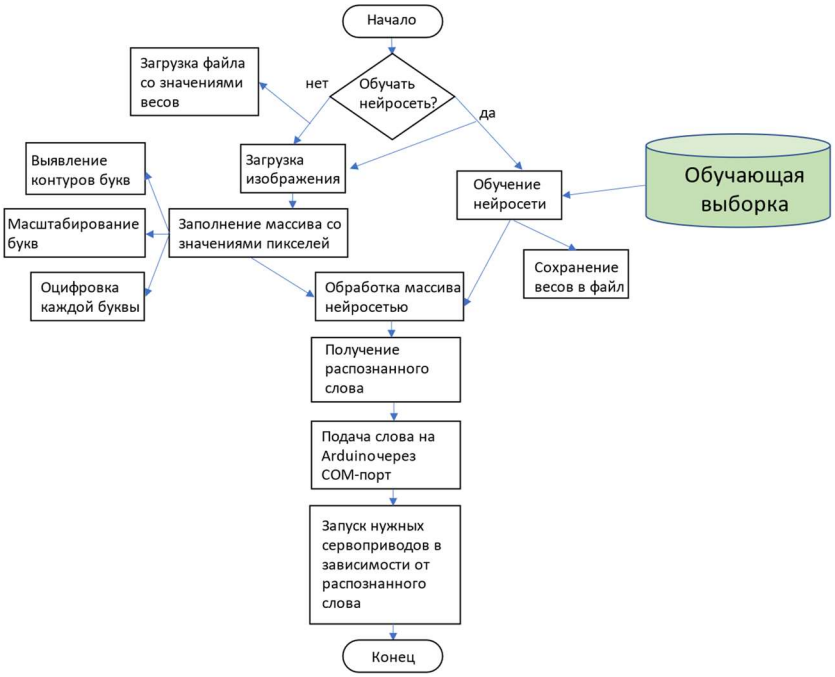


Рисунок 18 – Функциональная схема модели

ЗАКЛЮЧЕНИЕ

В данном проекте была создана собственная нейронная сеть, для которой было выполнено обучение на большом наборе букв в различных очертаниях. Нейросеть способна распознавать текст с изображения, делить его на буквы, определяя для каждой буквы соответствующий текстовый аналог.

Кроме того, был разработан макет устройства, на вход которого поступает сигнал в виде распознанной нейросетью буквы, и устройство приводит в движение соответствующие этой букве элементы согласно азбуке Брайля.

Для слабовидящих и незрячих людей данный инструмент будет очень удобен в использовании. Устройству можно передавать изображения текста, или текстовые файлы с информацией, а оно в свою очередь интерпретирует содержимое файлов в тактильный шрифт. В перспективе будет создана итоговая панель с определенным количеством автоматизированных ячеек Брайля, будет настроена передача файлов на устройство сразу несколькими способами, например, в данный момент идет добавление еще передачи через Bluetooth.

СПИСОК ИСПОЛЬЗУЕМЫХ ИСТОЧНИКОВ

1. Нейронные сети для начинающих. Часть 1 [Электронный ресурс] URL: https://habr.com/ru/articles/312450/ (дата обращения: 06.12.2023)

2. Нейронные сети для начинающих. Часть 2 [Электронный ресурс] URL: https://habr.com/ru/articles/313216/ (дата обращения: 13.12.2023)

3. Градиентный спуск: всё, что нужно знать [Электронный ресурс] URL: https://neurohive.io/ru/osnovy-data-science/gradient-descent/ (дата обращения: 10.11.2023)

Приложение А. Отзыв

Федеральное государственное
бюджетное образовательное учреждение
высшего образования
«Национальный исследовательский
университет «МЭИ» (ФГБОУ «МЭИ»)
111250, г. Москва,
вн.тер.г. муниципальный округ Лефортово,
ул. Красноказарменная, д. 14, стр. 1
Тел.: (495) 362-75-60, факс: (495) 362-89-38
E-mail: universe@mpei.ac.ru
http://www.mpei.ac.ru

№ _____

«___» _____ 20 __ г.

ОТЗЫВ

на **проектную** работу
(проектную / исследовательскую)

по теме **«Нейросеть для распознавания текста на изображении и его интерпретация для людей с ограниченными возможностями»**
(наименование работы)

учащегося ___**10**_____ класса _____ **ГБОУ Школа №1155** _____
(класс) *(наименование образовательной организации)*

_____ **Казаков Иван Сергеевич** _____.
(фамилия, имя, отчество (при наличии), заполняется на каждого участника)

В проектной работе «Нейросеть для распознавания текста на изображении и его интерпретация для людей с ограниченными возможностями» автор предлагает новое средство чтения текстовой информации для незрячих и плохо видящих людей. В проекте создана нейронная сеть, которая распознает печатный текст с загруженного изображения. Далее этот текст подается на устройство, которое представляет собой панель с выдвижными элементами согласно шрифту Брайля для тактильного чтения.

В работе приведено подробное описание созданной программы, изложены этапы сборки демонстрационного макета устройства, приведены качественные схемы и иллюстрации, подтверждающие работоспособность созданного продукта. Автором продемонстрирован высокий уровень владения выбранным языком программирования, умение создавать прототипы электронных устройств, способность ставить перед собой актуальные социально-значимые задачи и решать их с помощью перечисленных навыков.

Цели работы автором достигнуты, актуальность представленных в работе достижений и их практическая значимость подтверждены. Функционал разработанного устройства удобен пользователю и в перспективе может найти широкое применение в жизни.

ст. преподаватель _____ Дружинин А.А.
(должность) *(подпись)* *(И. О. Фамилия)*

М. П.