Государственное бюджетное общеобразовательное учреждение

Школа №1770

Проектная работа

**МЕТОД ОРИЕНТАЦИИ В ПРОСТРАНСТВЕ С ПРИМЕНЕНИЕМ КОМПЬЮТЕРНОГО ЗРЕНИЯ ДЛЯ СЛЕПЫХ И СЛАБОВИДЯЩИХ ЛЮДЕЙ**

Работу выполнили:

**Турлычкин** Егор Владимирович (10 класс «А») –

учащийся 10 класса «А»

ГБОУ Школа №1770

Научный руководитель:

**Епифанцев** **Сергей** Владимирович

учитель информатики

ГБОУ Школа №1770

Москва 2019

Оглавление

 [ВВЕДЕНИЕ 1](#_Toc5534179)

[Актуальность 1](#_Toc5534180)

[Проблема 1](#_Toc5534181)

[Цель проекта 1](#_Toc5534182)

[Задачи проекта 1](#_Toc5534183)

[Методы, применяемые в работе 2](#_Toc5534184)

[Ожидаемые результаты проекта 2](#_Toc5534185)

[Возможные риски проекта 2](#_Toc5534186)

[1. Теоретическое ознакомление с технологии создания датчиков 3](#_Toc5534187)

[1.1. Программирование микроконтроллеров 3](#_Toc5534188)

[1.2. Выбор микропроцессора 3](#_Toc5534189)

[1.3. Изготовление плат 3](#_Toc5534190)

[1.4. Код для микроконтроллера 3](#_Toc5534191)

[2. Создание прототипа устройства 4](#_Toc5534192)

[2.1. Прошивка микроконтроллера 4](#_Toc5534193)

[2.2. Вывод информации с датчиков 4](#_Toc5534194)

[2.3. Программное обеспечение 4](#_Toc5534195)

[2.4. Печать кейса для датчика 4](#_Toc5534196)

[3. Разработка одежды 5](#_Toc5534197)

[3.1. Виртуальная визуализация 5](#_Toc5534198)

[3.2. Раскрой ткани. Прострочка 5](#_Toc5534199)

[4. Компьютерное зрение 5](#_Toc5534200)

[4.1. Установка TensorFlow-GPU 6](#_Toc5534201)

[4.2. Настройка каталога TensorFlow и виртуальной среды Anaconda 6](#_Toc5534202)

[4.3. Сбор изображений 6](#_Toc5534203)

[4.4. Обработка изображений 7](#_Toc5534204)

[4.5. Создание меток 7](#_Toc5534205)

[4.5. Запуск тренировки 7](#_Toc5534206)

[4.5. Завершение тренировки 8](#_Toc5534207)

[Результаты 8](#_Toc5534208)

[Дорожная карта 9](#_Toc5534209)

[Используемые источники 10](#_Toc5534210)

# ВВЕДЕНИЕ

## Актуальность

По статистике ВОЗ (Всемирная организация здравоохранения) в мире проживает 36 миллионов человек, поверженных слепотой. По последним подсчетам, приблизительно 1,3 млрд человек в мире живут с той или иной формой нарушения зрения. За последнее время в мире бурно развивается новая прикладная область математики, специализирующаяся на искусственных нейронных сетях, на базе которой мы реализуем применение «компьютерного зрения». Актуальность исследований в этом направлении подтверждается массовостью различных применений современных технологий, в том числе использующих нейросеть. Кроме того, как на отечественном, так и на зарубежном рынке ассистивных устройств отсутствуют современные средства и методики реабилитации для слепых и слабовидящих.

## Проблема

Отсутствие современных средств реабилитации, позволяющих свободно ориентироваться в пространстве людям с ОВЗ по зрению, как на отечественном, так и на международном рынке.

## Цель проекта

Создание ассистивного устройства для слепых и слабовидящих людей, позволяющего не только свободно ориентироваться в пространстве, но и удовлетворить потребности в социальной мобильности и независимости людей с ограниченными возможностями здоровья (ОВЗ).

## Задачи проекта

1. Изучение принципов создания одежды, работа с программой CLO.
2. Изучение технологий сборки устройств на базе микроконтроллеров Arduino.
3. Изучение технологии машинного обучения (TensorFlow).
4. Разработка стратегии создания прототипа.
5. Сборка ультразвукового датчика.
6. Применение нейронных сетей на практике.
7. Нахождение способа встраивания датчиков и проводов электроснабжения в одежду, не сковывающие передвижения и деятельность потребителя.
8. Определить материал для одежды, который будет удобный при ежедневном ношении.
9. Закупка оборудования и расходных материалов.
10. Создание интересного образа. Разработать дизайн одежды, который будет не только максимально выполнять свои функции, помогая свободно ориентироваться в пространстве, но еще и модно выглядеть.

## Методы, применяемые в работе

1. Робототехника, в том числе пайка микроконтроллеров.
2. Программирование микроконтроллеров.
3. Реализация устройства.
4. Анализ, синтез и обобщение данной работы.

## Ожидаемые результаты проекта

Разработка специализированной одежды для слепых, которая облегчит ориентирование в пространстве людей с ОВЗ по зрению.

## Возможные риски проекта

Кроме непреднамеренной ошибки, возможна ситуации целенаправленных атак по дискредитации систем распознавания.

В области машинного обучения, около 80% времени уходит на подготовку данных.

## Ожидаемые результаты проекта

* «Ультразвуковая насадка на трость».

Разработка состоит из ультразвукового датчика, платы Arduino и крепежа для насадки на трость.

Недостатки: Один датчик не дает достаточного количества информации и реагирует только на объекты уровня трости.

* «Ультразвуковая наручная трость для слепых».

Разработка состоит их тех же деталей, что и предыдущая, за исключением крепежа. В этой разработке он крепится на руке.

Недостатки: прибор не оставляет места для привычной трости, но не дает полной информации о кочках, ямках и других объектах на земле. Ограничение в получении информации.

* «Ультразвуковой бейдж».

Разработка состоит из ультразвукового датчика, Arduino Uno, ремешка для ношения на шее.

Недостатки: те же, что и у «Ультразвуковой насадки на трости», то есть информации с единственного датчика недостаточно.

# 1. Теоретическое ознакомление с технологии создания датчиков

## Программирование микроконтроллеров

Микроконтроллер – это компьютерный процессор, который признает бинарные данные как входной сигнал, обрабатывает его согласно инструкциям, которые хранятся в памяти, а далее обеспечивают выход. Микроконтроллер, как и остальные компоненты датчика напаивается на плату. Далее в микроконтроллер зашивается программа, которая запрашивает входные данные, выполняет условия, анализ данных и выводит информацию в любом доступном формате (звуко- или вибросигнал).

## Выбор микропроцессора

За, почти, пол века существования микроконтроллеров их разнообразие и применение оказалось в большом количестве. Для наших целей нужно было выбрать контроллер, способный функционировать в системе с сонаром и остальными компонентами при напряжении от обычной пальчиковой батарейки или от Power Banka (~3,6V).

На данный момент лидерами в продаже Микроконтроллеров являются Intel, microchip, Atmel, Motorola и многие другие компании.

Простейшие микроконтроллеры применяются в случаях, когда не требуется высокая производительность, поэтому нами был выбран контроллер компании Atmel. Наилучшим образом подходят для решения поставленной задачи. Кроме того, они компактны, то есть занимают мало места плате и не имеют лишних ног (пинов), что позволит на максимум использовать его функционал.

## Изготовление плат

Способов изготовления плат достаточно много, мы опробовали каждый. С начала нам предстояло спроектировать и нарисовать расположение компонентов и контактов на плате. Для этого мы воспользовались программой EAGLE от Autodesk. Изначально мы рисовали разводку прямо на текстолите, с помощью тонкой кисточки и лака.

Изготовление печатной платы – долгий и сложный процесс, который можно произвести в домашних условиях.

Кроме того, можно заказать изготовление платы на заводе – это тоже длительный процесс.

Мы же остановились на варианте гравирования печатной платы. Имея специальный граверный аппарат, гравирование небольшой платы – подобно нашей, занимает порядка 10-15 минут.

## Код для микроконтроллера

Программный код мы писали на языке c++ с ассемблерными вставками, для этого использовали программу Code Vision AVR.

Но в связи с устареванием данного ПО, мы перешли на Arduino IDE. Далее, с помощью программатора (или черед Arduino Uno) запрограммировали микропроцессор.

# 2. Создание прототипа устройства

## 2.1. Прошивка микроконтроллера

Для прошивки микроконтроллера мы интегрировали встроенный программатор, поэтому прошить микропроцессор можно через USB порт на компьютере. Установив необходимые драйвера, нужно запустить программу для прошивки микроконтроллера. Далее проверяем связь с процессором, при подтверждении – готовности к работе, начинаем процесс записи ОС на микропроцессор нашего устройства. Для этого указываем путь .Hex файла и начинаем процесс автоматической записи. После записи микроконтроллер произведет перезагрузку и, при получении питания, начнет свою работу.

## 2.2. Вывод информации с датчиков

При планировании проекта, в качестве вывода информации было предложено использовать звуковые сигналы, по аналогии с машинным парктроником. Но в процессе апробаций разработали тестовую плату и код под плату с вибромотором. Далее было решено использовать именно этот метод вывода информации, так как он позволяет охватить больший круг потребителей нашего продукта.

## 2.3. Программное обеспечение

Информация со всех датчиков будет поступать в приложение на телефоне, где обрабатывается и выводит информацию доступным сигналом пользователя нашего устройства.

## 2.4. Печать кейса для датчика

В качестве программного обеспечения для визуализации мы использовали Fusion 360 от Autodesk и Cura. В Fusion 360 разработали 3D-модель, опираясь на имеющиеся особенности датчика. В Cura мы переводили модели в формат для 3D-принтеров. Имея базовые навыки работы с 3D-принтером – распечатали модель

# 3. **Разработка одежды**

## 3.1. Выбор дизайна для одежды Так как для людей с ОВЗ по зрению важен внешний вид одежды, перед началом моделирования одежды, необходимо выбрать стиль и дизайн одежды. Мы придерживались урбанизма и аскетизма для повышения функциональности одежды. Как дизайн, мы выбрали цвета Pantone 2019. Также было необходимо разработать систему для скрытого внедрения датчиков и их удобного расположения, чтобы не пострадала функциональность. Мы решили использовать два тонких слоя ткани, для проведения проводов и специальные кармашки для датчиков

## 3.2. Виртуальная визуализация

Для того, чтобы создать рисунок нам были нужны лекала в виде .jpeg картинки. Загрузили модель реального человека с помощью скана. Реальная ткань, свойства которой мы будем ввели в программу. Также мы провели измерения индивидуальной фигуры (мерки), и сделали 3Д скан человека. Оценка затрачиваемого материала на пошив.

## 3.3. Раскрой ткани. Прострочка

Перед раскроем ткани материал требуется проверить на специальном станке. Если брак точечный, то ткань можно использовать, а бракованные участки вырезать. Если брак масштабный, например, полоса вдоль рулона, то тогда ткань возвращают производителю. Такие рулоны часто сложно использовать в массовом производстве, но если это небольшая партия, как у нас, то материал можно оставить. Далее идет раскрой. Вообще, раскрой — самый занимательный этап пошива одежды. В крупных компаниях используют специальные станки, которые автоматически настилают ткани и вырезают нужную область. Технологический процесс изготовления одежды (технология шитья) — это изменение формы, размеров, частично и физических свойств обрабатываемых материалов (деталей, узлов) при соблюдении определенной последовательности выполнения операций с целью придания одежде необходимой объемной формы. При шитье и происходит соединение отдельных деталей и узлов с целью получения готового изделия, влажно-тепловую обработку для придания изделию необходимого объема, формы и товарного вида.

# 4. Компьютерное зрение

По мере изучения данной ниши – компьютерного зрения – мы выявили ряд причин, по которым нами было принято решение использовать технологию машинного обучения TensorFlow. TensorFlow – программная библиотека с открытым исходным кодом.

Далее нашей задачей являлось изучение руководств по использованию TensorFlow для обучение собственной нейросети обнаружения объектов. Для этого мы:

## 4.1. Установка TensorFlow-GPU

Скачали, установили и настроили Anaconda3 с Python3.6.
Благодаря команде pip install -- upgrade tensorflow-gpu загрузили последнюю версию TF. Далее импортировали TF в Python import tensorflow as tf

## 4.2. Настройка каталога TensorFlow и виртуальной среды Anaconda

Для запуска или обучения модели обнаружения объектов потребуется использовать дополнительные пакеты Python, специальных дополнений к переменным PATH и PYTHONPATH.

C:\> conda create -n tensorflow1 pip python=3.5

Затем активируйте среду, выпуская:

C:\> activate tensorflow1

Установите TensorFlow-gpu в этой среде:

(tensorflow1) C:\> pip install --ignore-installed --upgrade tensorflow-gpu

Установите другие необходимые пакеты, выполнив следующие команды:

(tensorflow1) C:\> conda install -c anaconda protobuf

(tensorflow1) C:\> pip install pillow

(tensorflow1) C:\> pip install lxml

(tensorflow1) C:\> pip install Cython

(tensorflow1) C:\> pip install jupyter

(tensorflow1) C:\> pip install matplotlib

(tensorflow1) C:\> pip install pandas

(tensorflow1) C:\> pip install opencv-python

(Примечание: пакеты "pandas" и "opencv-python" не нужны TensorFlow, но они используются в скриптах Python для создания TFRecords и для работы с изображениями, видео и веб-камера каналы.)

## 4.3. Сбор изображений

Для надежного определения объектов, для TensorFlow нужно подготовить большое количество изображений объекта чтобы натренировать классификатор обнаружения. Причем для тренировки надежного классификатора, «обучающие» изображения должны иметь случайные объекты вместе с нужным объектом и должны иметь различный фон и освещение. То есть должны быть изображения, где наш объект частично затеняется, перекрывается другим объектом и т.д.

Для каждого объекта мы сделали по 150-300 различных фотографий различного ракурса. При чем фотографировали, не прибегая к профессиональным камерам, а на обычный смартфон. Стоит отметить, что в добавок можно использовать готовые подборки.

## 4.4. Обработка изображений

После фотосъемки следует уменьшить качество фотографий, это позволит сэкономить время на тренировке классификатора.

Далее следует пометить объект на каждом изображении. Для этого следует скачать программу LabelImg. LabelImg - отличный инструмент для маркировки изображений, им мы и должны воспользоваться.

## 4.5. Создание меток

Последнее, что нужно сделать перед обучение – это создать метки. Имена классов сопоставляются с номерами идентификаторов классов, то есть карта меток сообщает тренеру «что есть» данный объект. В текстовом редакторе печатаем: [3]

item {

 id: 4

 name: 'Egor\_people'

}

item {

 id: 5

 name: 'TV'

}

item {

 id: 8

 name: 'chair'

## 4.5. Запуск тренировки

Для того, чтобы начать обучение нужно в каталоге \object\_detection нужно ввести команду:

python train.py --logtostderr --train\_dir=training/ --pipeline\_config\_path=training/faster\_rcnn\_inception\_v2\_pets.config

Далее, если всё было верно настроено, TensorFlow инициализирует обучение.

На каждом шаге выводится сообщение с количеством loss и каждый раз это значение уменьшается. Тренировку модели стоит продолжить до тех пор, пока значение loss не дойдет до 0,05, что займет, приблизительно 40 000 шагов.

Кстати, ход выполнения задания можно просмотреть при помощи тензорной панели. Для этого нужно открыть Anaconda Prompt, активировать виртуальную среду tensorflow1 и выполнить команду:

(tensorflow1) C:\tensorflow1\models\research\object\_detection>tensorboard --logdir=training

Эта команда создаст веб-страницу на локальном компьютере, которую можно просмотреть через браузер. Страница TensorFlow содержит информацию и графики, которые отображают процесс обучения. Там же есть график loss, программа каждые 5 минут сохраняет контрольные точки.

## 4.5. Завершение тренировки

Завершить обучение можно нажав сочетание клавиш Ctrl и C, находясь в командной строке. Нужно создать замороженный граф вывода. Для этого в папке \ object\_detection, “XXXX” в “ model.ckpt-XXXX” меняем на наибольший шаг.

Далее требуется экспортировать классификатор обнаружения объектов. Приложение готово, можно запускать.

# Результаты

1. Сшит прототип одежды для ребенка 10 лет. С встроенными кармашками для датчиков.
2. Эффективная дальность «парктроника» 4 метра. Способ питания: любой источник питания на 5v (вольт) до 2а (ампер), например используя аккумулятор телефона, Power bank или обычная пальчиковая батарейка формата АА, её емкости хватит на 100 часов работы системы (всех датчиков, не учитывая потребление Смартфона)
3. Приложение для Компьютера под ОС Windows; оптимизированное приложение на Android. Программа определяет изображение с веб-камеры компьютера или смартфона.

# Дорожная карта

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | сентябрь – октябрь | ноябрь - декабрь | январь - февраль | март - апрель | Май - … |
| Изучение технологий сборки устройств на базе Arduino  | Лекция | Интернет источник [1] | Практика |  |  |
| Изучение технологии машинного обучения (TensorFlow) |  | Интернет источник [2] | Практика, руководство |  |  |
| Стратегия работы над проектом  | Разработка приблизительного плана работы над проектом | Корректировка плана | Опережение плана - корректировка | Апробация в Специальной коррекционной школе-интернат №2 Департамента образования г. Москвы | Исправление выявленных ошибок в ходе апробации |
| Применение нейронных сетей на практике |  | Установка софта, пробный запуск обучения модели | Сборка приложения на телефон | Более углубленное обучение большим количеством объектов  |  |
| Закупка оборудования и расходных материалов |  | Закупка микроконтроллеров, отладочных плат и соединительных проводов | Закупка паяльной станции, микроконтроллеров другого формата, плат, остальных компонентов |  |  |
| Разработка готового прототипа датчика | Установка нужного Софта и подключение библиотек | Программирование, отладка. Первый прототип датчика | Сборка 5 штук | Сборка датчика с применением новых компонентов |  |

# Используемые источники

1. Цикл статей «Цикл статей AVR» [электронный ресурс] <http://easyelectronics.ru/category/avr-uchebnyj-kurs>
2. Руководство TensorFlow [электронный ресурс]

<https://www.tensorflow.org/tutorials>

1. Открытый исходный код автора EdjeElectronics [электронный ресурс] <https://github.com>
2. Статья AVR: подключаем дальномер HC-SR04 к atmega8 [Электронный ресурс]

http://mainloop.ru/avr-atmega/avr-ultrasonic-sensor-hc-sr04.html

1. Белов А. В. Конструирование устройств на микроконтроллерах. [печатное издание]

Издательство: НИТ 2005 год

1. Литвак А. Г. «Психология слепых и слабовидящих» [печатное издание]

СПб. Издательство РГПУ, 1998 год