

Федеральное государственное автономное
образовательное учреждение
высшего образования
«СИБИРСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»

Институт космических и информационных технологий

институт

Вычислительная техника

кафедра

УТВЕРЖДАЮ
Заведующий кафедрой

_____ О. В. Непомнящий

подпись инициалы, фамилия

« _____ » _____ 2020 г.

МАГИСТЕРСКАЯ ДИССЕРТАЦИЯ

Интеллектуальная система управления змеевидным роботом

Тема

09.04.01 «Информатика и вычислительная техника»

код и наименование направления

09.04.01.06 «Микропроцессорные системы»

код и наименование магистерской программы

Научный

руководитель

подпись, дата

профессор, зав кафедры, к.т.н
должность, ученая степень

О.В. Непомнящий
инициалы, фамилия

Выпускник

подпись, дата

Е.А. Минин
инициалы, фамилия

Рецензент

подпись, дата

руководитель деп.инф.техн.
должность

А.С. Стаценко
инициалы, фамилия

Нормоконтролер

подпись, дата

О.В. Непомнящий
инициалы, фамилия

Красноярск 2020

Федеральное государственное автономное
образовательное учреждение
высшего образования
«СИБИРСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»

Космических и информационных технологий

институт

Вычислительная техника

кафедра

УТВЕРЖДАЮ

Заведующий кафедрой

_____ О. В. Непомнящий

подпись инициалы, фамилия

« ____ » _____ 2019 г.

ЗАДАНИЕ НА ВЫПУСКНУЮ КВАЛИФИКАЦИОННУЮ РАБОТУ
в форме _____ **магистерской диссертации**

Студенту Минину Евгению Анатольевичу
фамилия, имя, отчество

Группа КИ18-01-6М Направление (специальность) 09.04.01
номер код

«Информатика и вычислительная техника»
наименование

Тема выпускной квалификационной работы «Интеллектуальная система управления змеевидным роботом»

Утверждена приказом по университету № 19249/с от 21 декабря 2018

Руководитель ВКР Непомнящий О.В. профессор, зав. кафедры ВТ ИКИТ СФУ,
к.т.н.
инициалы, фамилия, должность, ученое звание

Исходные данные для ВКР разработка системы управления, функционирующей на основе модуля принятия решений, базирующегося на нейронной сети.

Перечень разделов ВКР Реферат, Введение, Анализ задания выпускной квалификационной работы, Разработка модели интеллектуального устройства управления змеевидным роботом, Разработка лабораторного макета змеевидного робота, Разработка алгоритмического и программного обеспечения, Заключение, Список сокращений, Список используемых источников.

Перечень графического материала Листинг кода программы Python, презентация, демонстрационное видео.

Руководитель ВКР _____
подпись

О.В. Непомнящий
инициалы, фамилия

Задание принял к исполнению _____
подпись

Е.А. Минин
инициалы, фамилия

« ____ » _____ 2019

РЕФЕРАТ

Выпускная квалификационная работа по теме «Интеллектуальная система управления змеевидным роботом» содержит в себе 65 страниц текстового документа, 31 использованных источников, 36 иллюстраций, 5 таблиц, 3 листинга программного обеспечения, 3 приложения.

НЕЙРОННАЯ СЕТЬ, ИНТЕЛЛЕКТУАЛЬНАЯ СИСТЕМА, РОБОТ, ДЕФЕКТ, РАСПОЗНАВАНИЕ, ИЗОБРАЖЕНИЕ.

– Целью проводимых в рамках магистерской диссертации исследований является разработка прототипа интеллектуальной системы управления для колесных секций змеевидного робота, предназначенного для выполнения поисковых работ в полостях трубопроводов. Для достижения поставленной цели определены следующие основные задачи ВКР:

– Определение цели и основных методов исследования. Анализ известных решений, выбор способов решения поставленных задач.

– Разработка модели интеллектуального устройства управления змеевидным роботом. Разработка перечня метрик моделирования. Моделирование устройства управления на основе нейронной сети и определение основных технических требований к прототипу змеевидного робота.

– Определение функционального состава для изготовления лабораторного макета робота. Разработка алгоритмического и программного обеспечения для лабораторного макета.

– Выбор элементов и узлов для реализации устройства управления. Изготовление и тестирование лабораторного макета. Отработка полученных технических решений и анализ результатов диссертационного исследования.

Во введении раскрывается актуальность работы, изложены цели и задачи работы.

В первой главе приведены результаты анализа существующих решений, а также результаты выбора методов и средств разработки.

Во второй главе изложены результаты разработки модели устройства управления, а также результаты моделирования системы.

В третьей главе описывается процесс разработки и прототипирования аппаратной части колесной секции и системы управления роботом.

В четвертой главе изложены результаты разработки алгоритмического и программного обеспечения, а также приведены данные отработки системы управления на изготовленном демонстрационном стенде.

В итоге выполнения ВКР разработан метод интеллектуального управления и архитектура системы. Выполнено программное и натурное моделирование. Изготовлен лабораторный стенд, при помощи которого проверена работоспособность системы. Полученные результаты показали полное соответствие модели и прототипа работа поставленной цели.

СОДЕРЖАНИЕ

ВВЕДЕНИЕ	4
1 Анализ задания выпускной квалификационной работы	6
1.1 Обзор известных аналогов	6
1.1.1 Змеевидный робот «ЗМЕЕЛОК 3»	6
1.1.2 Система телеинспекции Кроулер S300 (Великобритания) робот ...	8
1.1.3 Робот для телеинспекции трубопроводов S-100.....	10
1.2 Выводы	12
2 Разработка модели интеллектуального устройства управления змеевидным роботом.....	13
2.1 Функциональный состав устройства управления змеевидным роботом на основе нейронной сети	13
2.2 Общие принципы и алгоритмы функционирования	14
2.3 Выбор интеллектуальной системы.....	15
2.4 Выбор нейронной сети для распознавания изображений.....	16
2.5 Выбор средств моделирования	17
2.5.1 Amazon Rekognition	18
2.5.2 Teachable Machine	19
2.5.3 Библиотека TensorFlow	20
2.6 Выбор метрик моделирования.....	21
2.7 Обучение нейронной сети	23
2.8 Моделирование и анализ полученных результатов	26
2.9 Выводы.....	29
3 Разработка лабораторного макета змеевидного робота	31
3.1 Выбор компонентов для разработки лабораторного макета змеевидного робота	31
3.1.1 Выбора микропроцессорного ядра.....	31
3.1.2 Выбор видеокамеры для одноплатного компьютера	33
3.1.3 Выбор датчика препятствий.....	34

3.1.4	Выбор мотора и колес для передвижения робота.....	35
3.1.5	Выбор платформы для крепления компонентов.....	36
3.1.6	Выбор сервоприводов для поворота камеры	37
3.2	Сборка лабораторного макета.....	38
3.3	Выводы.....	40
4	Разработка алгоритмического и программного обеспечения.....	41
4.1	Разработка алгоритмов управления	41
4.2	Разработка программного обеспечения.....	43
4.2.1	Разработка программного обеспечения движения робота	44
4.2.2	Разработка программного обеспечения обнаружения дефектов ..	45
4.3	Проведение экспериментов.....	47
4.3.1	Эксперимент с дефектом в виде наличия камня в трубе	48
4.3.2	Эксперимент с дефектом в виде наличия отверстия в трубе	50
4.3.3	Эксперимент с отсутствием дефектов	51
4.4	Выводы.....	53
	ЗАКЛЮЧЕНИЕ	55
	СПИСОК СОКРАЩЕНИЙ.....	57
	СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ	58
	ПРИЛОЖЕНИЕ А	62
	ПРИЛОЖЕНИЕ Б.....	63
	ПРИЛОЖЕНИЕ В	64

ВВЕДЕНИЕ

Необходимость поддержания трубопроводов в рабочем состоянии заставляет искать новые эффективные методы контроля труб с целью выявления дефектов и трещин, а также коррозии на внешних и внутренних поверхностях.

В настоящее время в России эксплуатируются устаревшие нефтегазопроводы. Например, по статистике «Газпром» за 2013 год, 56529 тыс. км трубопроводов эксплуатируются более 20 лет, 41703 тыс. км более 30 лет, 19701 тыс. км более 40 лет и 9861 тыс. км более 50 лет. Ежегодно «Газпром» проводит капитальный ремонт отдельных участков газопроводов, этот метод ремонта с каждым годом становится все менее эффективней, так как общая протяженность газопроводов прогрессивно увеличивается [1].

На стенках технологических трубопроводов со временем образуются отложения. Этот процесс не зависит от способа эксплуатации системы и приводит к снижению её пропускной способности. Именно по этой причине возникают аварийные ситуации. Для минимизации риска порчи оборудования и локализации проблемных участков, проводится диагностика трубопроводов. Ниже перечислены основные методы выполнения этой процедуры без демонтажа и вскрытия оборудования:

Магнитооптическая дефектоскопия – позволяет обнаружить с помощью магнитного потока дефекты, присутствующие в ферромагнитном материале. Однако определить с достаточной точностью их глубину данным способом не представляется возможным [2].

Ультразвуковая диагностика – данным методом проверяется качество соединения компонентов трубопроводов, работающих под высоким давлением и на АЭС. В принцип обнаружения дефектов заложена способность волн ультразвукового диапазона легко проникать сквозь однородный материал. При наличии препятствий волны отражаются [3].

Опрессовка повышенным давлением – в данном случае инертные газы, газовая смесь или водяной пар нагнетаются в трубопровод так, чтобы создать внутри него давление, в 5 раз превышающее рабочее. Затем производится осмотр стыков, швов и мест соединения котельного оборудования и труб. Определение участков, в которых происходят утечки пара, осуществляется по наличию на них конденсата [4].

Видеодиагностика (теледиагностика) позволяет визуально оценить состояние трубопровода. Для анализа используются видеокамеры, смонтированные на проталкиваемом стеклопластиковом прутке или на роботах. Роботы, перемещаясь внутри трубопровода, снимают всё, что встретят на своём пути. Эта техника способна выявить грубые нарушения целостности труб, протечки на сегментах в грунтах или закрытых тоннелях, места образования крупных засоров и илистых отложений. Такую методику приняли на вооружение многие профильные строительные компании, поэтому данный метод диагностирования заслуживает отдельного внимания [5].

Видеодиагностика трубопроводов является одним из самых современных способов проведения инспекции внутренней поверхности труб. На сегодняшний день данный метод может проводиться как в ходе планового осмотра системы, так и при аварийных ситуациях. Используя видеодиагностику, удастся выявить возможные повреждения, поломки, засоры, переломы труб и слабые места трубопровода. При этом отсутствует необходимость в демонтаже труб. Проведение данного осмотра предполагает использование специализированной роботизированной техники [6].

1 Анализ задания выпускной квалификационной работы

1.1 Обзор известных аналогов

Ранее были рассмотрены некоторые способы диагностики трубопроводов без использования роботизированной техники. Однако, для более эффективной инспекции используют специальную роботизированную технику.

Рассмотрим более подробно несколько таких устройств.

1.1.1 Змеевидный робот «ЗМЕЕЛОК 3»

Данный змеевидный робот был разработан в Санкт-Петербурге Центральным научно-исследовательским и опытно-конструкторским институтом робототехники и технической кибернетики (ЦНИИ РТК). «ЗМЕЕЛОК 3» может передвигаться по болоту, песку, ползать по внутренним поверхностям труб, может быть оснащен камерой, микрофоном и другими необходимыми устройствами [7]. На рисунке 1 изображен внешний вид робота.



Рисунок 1 – Внешний вид робота «ЗМЕЕЛОК 3»

Одной из особенностей конструкции данного змеевидного робота является возможность создания, на основе унифицированного мехатронного модуля-шарнира механической структуры, способной осуществлять локомоции и манипуляции. Конструкция модуля представляет собой двухстепенной активный шарнир с пересекающимися взаимно перпендикулярными осями, находящимися в одной плоскости. Вращение приводного зубчатого колеса активных вилок осуществляется серводвигателем Hitech 7940TH. В каждом таком звене стоит микроконтроллер Atmel ATmega8a, который содержит уникальный идентификатор, определяющий положение модуля в последовательной цепи. Управление осуществляется посредством интерфейса RS-485. Встроенное программное обеспечение блока управления сервоприводом вращения оси мехатронного модуля выполняет функции инициализации микроконтроллеров после подачи напряжения питания от внешнего источника и обработки поступающих по интерфейсу RS485 команд задания, запроса, считывания и передачи значения угла поворота выходного вала сервопривода относительно нейтрального положения.

Технические характеристики змеевидного робота «ЗМЕЕЛОК 3» представлены в таблице 1.

Таблица 1 – Технические характеристики «ЗМЕЕЛОК 3»

Параметр	Значение
Количество модулей, шт	2-15
Масса модуля, кг	0,35
Масса, кг	0,7-5,25
Длина модуля, мм	80
Диаметр, мм	80
Максимальный момент на каждой оси модуля, кН*мм	3,2
Количество сервоприводов, шт	4-30
Напряжение питания, В	40
Связь с ПК	USB
Электропитания	Внешний источник

Результаты экспериментов показывают, что разработанная модель змеевидного робота «ЗМЕЕЛОК-3» имеет существенно лучшие силомоментные и скоростные показатели по сравнению с ранними версиями. Однако повышенное энергопотребление для их реализации делает невозможным реализацию автономного по питанию ЗР с достигнутыми показателями. Модель используется для отработки новых режимов движения и алгоритмов управления с целью выработки правильных конструктивных и программно-аппаратных решений при создании специальных мобильных роботов змеевидного типа [8].

Существенный недостаток системы - отсутствие автономного питания, что не позволит исследовать трубопроводы на длинные дистанции.

1.1.2 Система телеинспекции Кроулер S300 (Великобритания) робот

Данная портативная видеоинспекция на колесном ходу S300 предназначена для обследования трубопроводов диаметром от 150 мм до 1000 мм и длиной до 250 м в зависимости от длины подключаемого кабеля. На сегодняшний день это одно из самых высокотехнологичных устройств подобного типа, представленных на рынке. В зависимости от комплектации цена всей системы колеблется около 2 500 000-3 000 000 руб. Внешний вид робота изображен на рисунке 2.



Рисунок 2 – Робот Кроулер S300

Для того чтобы привести видеоинспекцию в рабочее положение, требуется несколько минут. Прочное, износостойкое покрытие из кевларовых нитей защищает кабель от абразивного износа. Кабель способен воспринимать значительные механические воздействия, не теряя при этом работоспособности. В комплект поставки входит барабан, что исключает переломы и запутывания. Барабан снабжен счетчиком длины кабеля. Устройство поставляется с кабелем длиной 200 или 250 метров. Основная часть устройства – водонепроницаемая CMOS камера – крепится непосредственно на тележке. Она обеспечивает передачу высококонтрастного изображения обследуемых областей на 10,4-дюймовый TFT-монитор. Объектив камеры защищен сапфировым стеклом, устойчивым к царапинам. Корпус камеры изготовлен из стали с антикоррозионным покрытием. Камера полностью водонепроницаема и может выдерживать давление до 5,5 бар. Укомплектована светодиодами последнего поколения с куполообразными линзами, обеспечивающими хорошее, равномерное освещение зон инспекции. Визуальный контроль осуществляется при помощи TFT-монитора, помещенного в кейс из ударопрочного пластика. Результаты исследования в формате MPEG2 можно сохранить на жесткий диск или USB накопитель. Используемое программное обеспечение совместимо с WinCam и позволяет также вводить до шести страниц текстовой информации [9].

Технические характеристики робота Кроулер S300 представлены в таблице 2.

Таблица 2 – Технические характеристики робота Кроулер S300

Параметр	Значение
Разрешение камеры	720x576
Вес камеры	1,2 кг
Освещение камеры	светодиодное
Питание тележки	Постоянный ток, 48 В
Количество колес	4 шт.

Окончание таблицы 2

Пятискоростной электродвигатель	150 Вт
Вес тележки в макс. снаряжении	40 кг
Питания БУ	Переменный ток, 220 В
Монитор	10,4 дюйма, TFT
Разрешение экрана	800x600
Формат видео	MPEG2
Хранение данных	жесткий диск, USB накопитель
Вес БУ	13 кг
Длина кабеля	200-250 м
Диаметр кабеля	9,6 мм
Конструкция барабана	Сталь
Вес кабеля и барабана	50 кг

Однако робот не мобилен и не может исследовать трубы протяженностью более 250 м. При этом общий вес системы составляет около 100 кг., что затрудняет транспортировку. Кроме того, система обладает довольно высокой стоимостью.

1.1.3 Робот для телеинспекции трубопроводов S-100

Робот S-100 применяется для инспекции водопровода, канализации, вакуумных систем, сантехнических сооружений. Робот позволяет выявить как крупные дефекты (проломы, расстыковка труб), так и детальную картину каждого квадратного сантиметра трубы. В зависимости от комплектации цена всей системы колеблется в районе 2 500 000 руб. Внешний вид робота изображен на рисунке 3.



Рисунок 3 – Внешний вид робота S-100

Робот S-100 применяется для труб и каналов диаметром 100 – 600 мм. Ручной лифт камеры – дополнительная опция. С ручным лифтом камеры робот S-100 сможет исследовать трубы до 800 мм в диаметре. Температура использования: от -20 до + 55 °С.

Робот S-100 оснащен фронтальной поворотной камерой с автоуровнем. Класс влагозащиты робота IP68, S-100 может работать на глубине до 10 метров под водой. В стандартной комплектации оснащается ручной намоткой и размоткой кабеля. Электрическая намотка и размотка кабеля - дополнительная опция. Робот S-100 может оснащаться кабелем длиной до 120 метров [10].

Технические характеристики робот S-100 представлены в таблице 3.

Таблица 3 – Технические характеристики робота S-100

Параметр	Значение
Глубина просмотра	0-80 м
Общий вес	110 кг
Диаметр труб	100-800 мм
Цветной монитор	Да
Диагональ монитора	12,1”
Угол обзора камеры	120
Влагозащищенность	IP68
Функция записи	Да
Воспроизведений	Да
Питание	Батарея или 200 В
Врем работы от аккумулятора	6
Встроенный трансмиттер	Нет
Счетчик расстояния	Есть

Недостатки предыдущей модели в полной мере можно отнести и данному роботу.

1.2 Выводы

На основании результатов анализа известных решений в предметной области исследований определена основная цель ВКР и задачи, предполагаемые к решению для ее достижения.

Рассмотрен ряд существующих аналогов, выявлены основные достоинства и недостатки современных решений. Определено, что в настоящее время наиболее распространенными являются роботы, обладающие достаточно высокими массогабаритными характеристиками и связанные с объектом управления посредством кабеля. Автономные роботы практически не применяются, известны единицы решений обладающие достаточно высокой стоимостью.

Таким образом, выявлена необходимость улучшения получаемых результатов инспекций трубопроводов. Для этого предложено увеличить степень автономности роботов-инспекторов. Определено, что этого можно достичь путем разработки новых интеллектуальных методик и технологий исследования технологических трубопроводов. Выявлено, что среди известных подходов наибольший интерес вызывает технологии видеодиагностики, поскольку такой подход позволяет выявить максимум возможных дефектов.

Это позволило перейти к разработке функционального состава интеллектуальной системы управления робота. Далее требуется выполнить обзор существующих методов и способов организации интеллектуального взаимодействия электронных систем УУ, а также выбрать компоненты и определить перечень основных узлов лабораторного демонстратора.

2 Разработка модели интеллектуального устройства управления змеевидным роботом

2.1 Функциональный состав устройства управления змеевидным роботом на основе нейронной сети

На рисунке 4 представлена разработанная общая функциональная схема прототипа устройства управления (УУ) ведущей секции змеевидного робота.

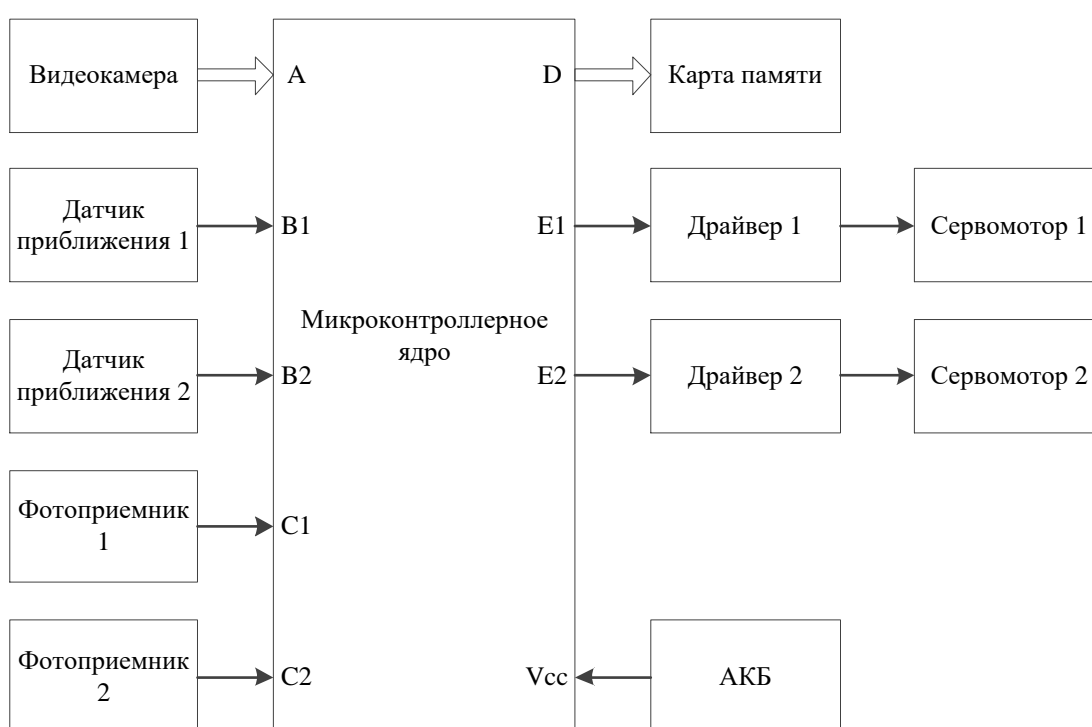


Рисунок 4 – Устройство управления

Основной составляющей УУ является микропроцессорное ядро (МЯ), при этом источником питания служит аккумуляторная батарея. Движение секций змеевидного робота осуществляются посредством двух сервомоторов, соединенных с помощью драйверов управления, которые подключены к МЯ. Основная часть системы – обученная нейронная сеть, которая функционирует под управлением МЯ, получает изображения от видеокамеры по шине,

анализирует его на наличие дефектов и, в случае обнаружения, записывает изображение на карту памяти. Датчики приближения контролируют наличие препятствий на пути робота в время движения, а фотоприемники, установленные на колесах, являются измерителями пройденного расстояния.

2.2 Общие принципы и алгоритмы функционирования

На рисунке 5 представлен общий алгоритм функционирования робота.

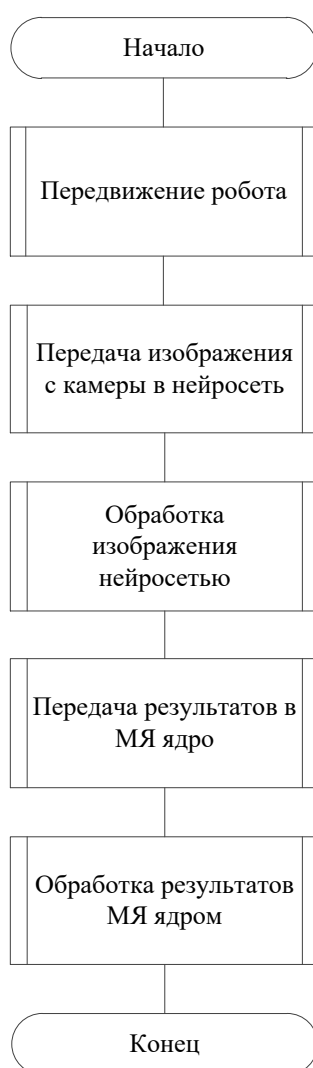


Рисунок 5 – Общий алгоритм функционирования

На основании вышеизложенного определены основные задачи УУ:

1. Считывание и передача изображений в центральную систему управления;
2. Обработка изображения нейронной сетью и формулирование результатов;
3. Передача результатов в МЯ для принятия решения;
4. Реализация протокола движения по трубе.

Так как моделируемая система управления роботом является автономной (т.е. функционирует без участия человека), решения принимаются самостоятельно. Для реализации интеллектуального модуля управления выполнен обзор известных методик в области создания самообучающихся систем. Определено, что наибольший эффект, в данном случае, следует ожидать от применения нейронной сети.

2.3 Выбор интеллектуальной системы

Известно, что под интеллектуальной системой (ИС) понимается объединенная информационным процессом совокупность технических средств и программного обеспечения, работающая во взаимодействии с человеком или автономно, способная на основании сведений об окружающей среде и собственном состоянии при наличии знаний и мотивации синтезировать цель, принимать решение о действии и находить рациональные способы достижения цели [11].

Для решения поставленных задач выделен ряд ИС:

- Системы с коммутативными способностями (с интеллектуальным интерфейсом, контекстные справочные системы, интеллектуальные базы данных);
- Экспертные системы (системы для решений сложных задач, многоагентные системы, доопределяющие системы);

- Самообучающиеся системы (системы, способные к самообучению, нейронные сети, информационные хранилища);
- Адаптивные системы (адаптивные информационные системы, CASE-технологии, компонентная технология) [12].

В данном случае, в состав самообучающейся системы входит нейронная сеть – это система соединенных и взаимодействующих между собой простых процессоров (искусственных нейронов), каждый процессор подобной сети имеет дело только с сигналами, которые он периодически получает, и сигналами, которые он периодически посылает другим процессорам [13].

Таким образом, основной составляющей интеллектуальной системы управления роботом является нейронная сеть, которая позволяет реализовать систему принятия решений при движении робота и выполнения поисковых задач в полостях труб.

В соответствии с заданием на выпускную квалификационную работу ниже рассмотрена отдельная часть функционирования нейронной сети, а именно обработка и принятие решений по результатам сканирования видеокамеры полости трубы. Реализация интеллектуальной системы управления полным функционалом робота на основе нейронной сети выходит за рамки ВКР.

2.4 Выбор нейронной сети для распознавания изображений

Для реализации моделирования распознавания изображений необходимо выбрать необходимый тип нейронной сети, использующийся для распознавания изображений. Результаты анализа предметной области позволили выделить ряд нейронных сетей, наиболее полно отвечающих требованиям задания на ВКР:

- Сверточная нейронная сеть – специальная архитектура нейронных сетей, предложенная Яном Лекуном, изначально нацеленная на эффективное

распознавание изображений [14]. Сверточные нейронные сети предназначены для обработки данных через несколько уровней массивов. Этот тип нейронных сетей используется в таких приложениях, как распознавание изображений или распознавание лиц. Основное различие между CNN и любой другой обычной нейронной сетью состоит в том, что CNN принимает входные данные в виде двумерного массива и работает непосредственно с изображениями, а не фокусируется на извлечении признаков, на котором сосредоточены другие нейронные сети [15].

- Рекуррентные нейронные сети – популярные модели, используемые в обработке естественного языка. Во-первых, они оценивают произвольные предложения на основе того, насколько часто они встречались в текстах. Это дает меру грамматической и семантической корректности. Такие модели используются в машинном переводе. Во-вторых, языковые модели генерируют новый текст. Обучение модели на поэмах Шекспира позволит генерировать новый текст, похожий на Шекспира. Рекуррентные нейронные сети называются рекуррентными, потому что они выполняют одну и ту же задачу для каждого элемента последовательности, причем выход зависит от предыдущих вычислений [16].

Основываясь на приведенных выше данных можно сделать вывод, что для реализации распознавания изображений лучше использовать сверточную нейронную сеть, так как именно она изначально нацелена на эффективное распознавание изображений.

2.5 Выбор средств моделирования

Моделирование позволяет наглядно увидеть, как создаваемая система будет распознавать объекты на изображениях, верно ли будет определять те или иные дефекты в трубопроводе. Определен ряд наиболее доступных программных средств, позволяющих выполнять моделирования и отработку

нейронных сетей для распознавания изображений: Amazon Rekognition, Teachable Machine и библиотека TensorFlow.

2.5.1 Amazon Rekognition

Amazon Rekognition – данный сервис позволяет автоматизировать анализ изображений и видео с помощью машинного обучения. Amazon Rekognition идентифицирует объекты, людей, текст, сцены и конкретные действия на изображениях и в видео, а также определяет неприемлемый контент. Позволяет с высокой точностью анализировать лица и открывает широкие возможности поиска лиц, которые можно применять для обнаружения, анализа и сравнений лиц в различных примерах использования, где необходима верификация пользователей, подсчет людей и обеспечение общественной безопасности. На рисунке 6 представлен пример идентификации объектов на изображении.

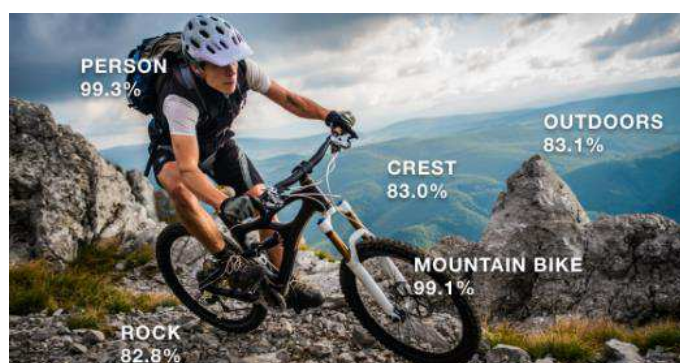


Рисунок 6 – Идентификация объектов Amazon Rekognition

Благодаря Amazon Rekognition Custom Labels можно идентифицировать на изображениях те объекты и сцены, которые соответствуют потребностям определенной сферы бизнеса. Например, можно построить модель для классификации конкретных деталей оборудования на сборочной линии или для выявления больных растений. Amazon Rekognition Custom Labels возьмет

на себя самую трудоемкую работу по разработке модели, поэтому предварительный опыт работы с машинным обучением не требуется. Нужно просто загрузить изображения объектов или сцен, которые вы хотите идентифицировать, а сервис позаботится обо всем остальном [17].

Не смотря на большое количество возможностей данного сервиса и его универсальности, он предлагает уровень бесплатного пользования с ограниченным функционалом. Так же нет гарантии, что данный сервис будет всегда доступен для использования.

2.5.2 Teachable Machine

Teachable Machine – это веб-инструмент, который делает создание моделей машинного обучения быстрым, простым и доступным для всех. В первой версии от 2017 года она позволяла тренировать нейросети на основе изображений и не поддерживала экспорт моделей. В версии от 2019 года Teachable Machine позволяет обучать компьютер распознавать собственные изображения, звуки и позы. Данный веб-инструмент можно научить классифицировать:

- Изображения с помощью файлов или веб-камеры;
- Звук, записывая короткие звуковые образцы;
- Положения тела, используя файлы или яркие позы в вашей веб-камере;
- И множество других классификаций.

На рисунке 7 показан пример реализации классификаций изображений, созданных с помощью Teachable Machine.

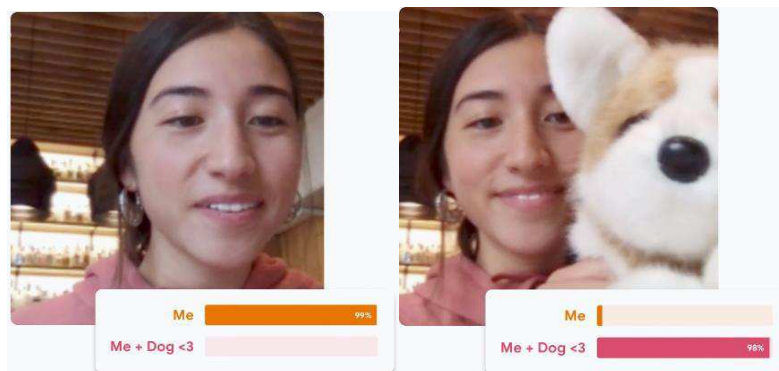


Рисунок 7 – Классификация изображения с помощью Teachable Machine

С помощью данного сервиса сделан крошечный сортировщик: эксперимент DIY, соединяющий Arduino и Teachable Machine. Стив Солинг использует Teachable Machine для общения по-новому, например, использует лицевые жесты для запуска звуков.

Модели, которые создаются с помощью Teachable Machine, являются реальными моделями Tensorflow.js, которые работают везде, где работает javascript, поэтому они прекрасно работают с такими инструментами, как Glitch, P5.js, Node.js и другими [18].

Данный веб-сервис является простым в понимании, в нем не сложно разобраться, но он так же может стать не доступным в процессе обучения нейронной сети.

2.5.3 Библиотека TensorFlow

TensorFlow – открытая программная библиотека для машинного обучения, разработанная компанией Google для решения задач построения и тренировки нейронной сети с целью автоматического нахождения и классификации образов, достигая качества человеческого восприятия [19].

Открыв исходный код библиотеки машинного обучения TensorFlow, в Google упростили процесс построения и развертывания сложных нейронных

сетей. TensorFlow предоставляет возможность использовать интерфейсы API для языков Python и C/C++, позволяющие подключаться к программе разработчика. TensorFlow заметно упрощают встраивание в приложения самообучаемых элементов и функций искусственного интеллекта, предназначенных для распознавания речи, организации компьютерного зрения или обработки естественного языка [20]. TensorFlow поддерживается в 64-битных системах, таких как Ubuntu 16.04 или более поздняя версия, Windows 7 или более поздняя версия, macOS 10.12.6 (Sierra) или новее (без поддержки графического процессора), Raspbian 9.0 или поздняя версия [21].

Исходя из предоставленных ранее способов обучения нейронных сетей, в качестве реализации моделирования была выбрана библиотека TensorFlow. Данная библиотека имеет открытый исходный код и не требует постоянного выхода в интернет для обучения нейронной сети. Поддерживает различные операционные системы, что упрощает реализацию моделирования.

2.6 Выбор метрик моделирования

В лабораторных условиях был создан макет трубы, который изображен на рисунке 8.



Рисунок 8 – Макет трубы

Известно множество разнообразных дефектов в трубопроводах. В рамках ВКР было решено выделить 5 ситуаций в которых может оказаться робот: 4 из которых являются результатами обнаружения различных типов дефектов, а одна ситуация без дефектов.

Для обучения нейронной сети было сделано несколько сотен изображений по каждому виду ситуаций, в качестве дефектов выступают: наличие камня в трубе, наличие какого-либо мусора, наличие отверстия в трубе и места отсоединения трубопроводов (стыки). На рисунке 9 слева-направо изображены дефекты: камень, мусор, отверстие и стык.

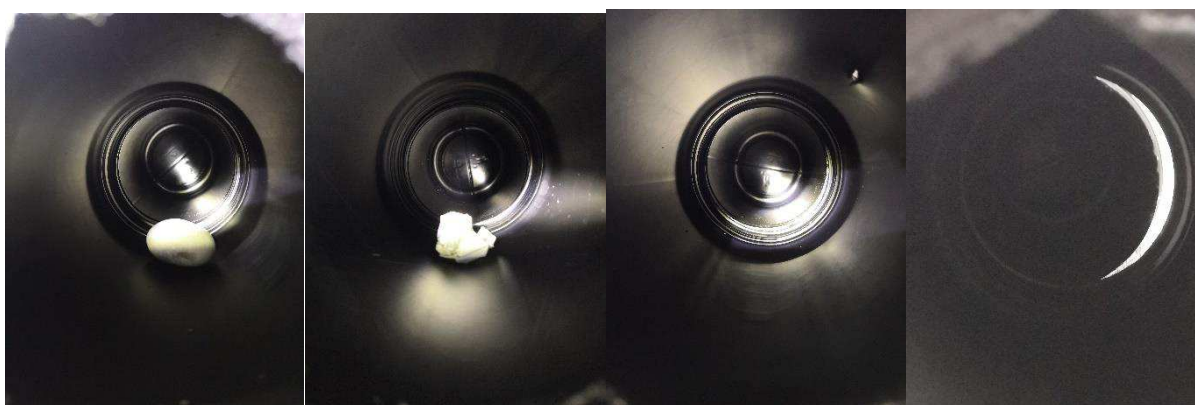


Рисунок 9 – Дефекты в трубопроводе

Также были подготовлены изображения трубы без дефектов (рисунок 10).



Рисунок 10 – Изображение трубы без дефектов

Определено, что результатом моделирования будет процентное соотношение вероятности идентификации дефекта по каждому виду.

2.7 Обучение нейронной сети

Обучение нейронной сети производилось на базе операционной системы Linux Mint версии 19.1 mate 64bit с использованием языка программирования Python и библиотеки TensorFlow.

В данной версии операционной системы уже был установлен Python 2.7.15, поэтому требовалась установка в терминале Linux только библиотеки TensorFlow с помощью следующих команд:

- `pip install --upgrade pip` – позволяет обновить менеджер пакетов Python, для стабильной установки дополнительных утилит и библиотек;
- `pip install tensorflow` – установка библиотеки TensorFlow.

Далее добавлен классификатор – это инструмент, который позволяет нейронной сети понимать, к чему относится неизвестный объект на изображении. Для использования классификатора был необходим набор изображений для обучения нейронной сети. В качестве изображений использовались фотографии демонстрационной трубы с наличием разного рода дефектов.

Классификатор и изображения находятся в одной папке под названием `pipe`. Внутри папки `pipe` находится папки:

- `scripts` – находятся исполняемые файлы классификатора на языке Python;
- `tf_file` – хранятся изображения и данные, которые создает нейронная сеть в процесс обучения;
- `defects_foto` – изображения дефектов трубопровода. Внутри данной папки находятся папки: `a_rock` (камень), `garbage` (мусор), `hole` (отверстие), `joint`

(стык), purely (чисто). В данных папках находятся различные варианты дефектов.

После этого был запущен процесс обучения нейронной сети, для анализа картинок из папки defects_foto. Для этого в терминале была прописана следующая команда:

```
python scripts/retrain.py --output_graph=tf_files/retrained_graph.pb --  
output_labels=tf_files/retrained_labels.txt --image_dir=tf_files/defects_foto
```

В данной команде указаны такие обозначения как:

- retrain.py – python-скрипт, который отвечает за запуск процесса обучения нейронной сети;

- output_graph – создаёт файл с графом данных. Будет использоваться для определения того, что находится на изображении;

- output_labels – создание файла с метками. В данном случае дефекты;

- image_dir – путь к папке, в которой находятся изображения с дефектами.

В начале обучения создаются текстовые файлы bottleneck, которые содержат компактную информацию об изображении, для лучшего считывания информации изображений для нейронной сети. На рисунке 11 показан процесс создания текстовых файлов bottleneck.

```
INFO:tensorflow:Creating bottleneck at /tmp/bottleneck/a_rock/IMG_20191220_191730_BURST010.jpg inception v3.txt
INFO:tensorflow:Creating bottleneck at /tmp/bottleneck/a_rock/IMG_20191220_191730_BURST027.jpg inception v3.txt
INFO:tensorflow:Creating bottleneck at /tmp/bottleneck/a_rock/IMG_20191220_191651_BURST048.jpg inception v3.txt
INFO:tensorflow:Creating bottleneck at /tmp/bottleneck/a_rock/IMG_20191220_191730_BURST011.jpg inception v3.txt
INFO:tensorflow:Creating bottleneck at /tmp/bottleneck/a_rock/IMG_20191220_191651_BURST039.jpg inception v3.txt
INFO:tensorflow:Creating bottleneck at /tmp/bottleneck/a_rock/IMG_20191220_191720_BURST033.jpg inception v3.txt
INFO:tensorflow:Creating bottleneck at /tmp/bottleneck/a_rock/IMG_20191220_191651_BURST010.jpg inception v3.txt
INFO:tensorflow:Creating bottleneck at /tmp/bottleneck/a_rock/IMG_20191220_191651_BURST043.jpg inception v3.txt
INFO:tensorflow:Creating bottleneck at /tmp/bottleneck/a_rock/IMG_20191220_191651_BURST027.jpg inception v3.txt
INFO:tensorflow:Creating bottleneck at /tmp/bottleneck/a_rock/IMG_20191220_191651_BURST041.jpg inception v3.txt
INFO:tensorflow:Creating bottleneck at /tmp/bottleneck/a_rock/IMG_20191220_191702_BURST040.jpg inception v3.txt
INFO:tensorflow:Creating bottleneck at /tmp/bottleneck/a_rock/IMG_20191220_191720_BURST041.jpg inception v3.txt
INFO:tensorflow:Creating bottleneck at /tmp/bottleneck/a_rock/IMG_20191220_191730_BURST037.jpg inception v3.txt
INFO:tensorflow:Creating bottleneck at /tmp/bottleneck/a_rock/IMG_20191220_191730_BURST012.jpg inception v3.txt
INFO:tensorflow:Creating bottleneck at /tmp/bottleneck/a_rock/IMG_20191220_191730_BURST035.jpg inception v3.txt
INFO:tensorflow:Creating bottleneck at /tmp/bottleneck/a_rock/IMG_20191220_191702_BURST001_COVER.jpg inception v3.txt
INFO:tensorflow:Creating bottleneck at /tmp/bottleneck/a_rock/IMG_20191220_191702_BURST033.jpg inception v3.txt
INFO:tensorflow:Creating bottleneck at /tmp/bottleneck/a_rock/IMG_20191220_191702_BURST013.jpg inception v3.txt
INFO:tensorflow:Creating bottleneck at /tmp/bottleneck/a_rock/IMG_20191220_191651_BURST006.jpg inception v3.txt
INFO:tensorflow:Creating bottleneck at /tmp/bottleneck/a_rock/IMG_20191220_191730_BURST002.jpg inception v3.txt
INFO:tensorflow:Creating bottleneck at /tmp/bottleneck/a_rock/IMG_20191220_191730_BURST019.jpg inception v3.txt
INFO:tensorflow:Creating bottleneck at /tmp/bottleneck/a_rock/IMG_20191220_191651_BURST044.jpg inception v3.txt
INFO:tensorflow:Creating bottleneck at /tmp/bottleneck/a_rock/IMG_20191220_191702_BURST008.jpg inception v3.txt
INFO:tensorflow:Creating bottleneck at /tmp/bottleneck/a_rock/IMG_20191220_191651_BURST001_COVER.jpg inception v3.txt
INFO:tensorflow:Creating bottleneck at /tmp/bottleneck/a_rock/IMG_20191220_191730_BURST036.jpg inception v3.txt
INFO:tensorflow:Creating bottleneck at /tmp/bottleneck/a_rock/IMG_20191220_191730_BURST008.jpg inception v3.txt
INFO:tensorflow:Creating bottleneck at /tmp/bottleneck/a_rock/IMG_20191220_191730_BURST020.jpg inception v3.txt
INFO:tensorflow:Creating bottleneck at /tmp/bottleneck/a_rock/IMG_20191220_191651_BURST017.jpg inception v3.txt
INFO:tensorflow:Creating bottleneck at /tmp/bottleneck/a_rock/IMG_20191220_191651_BURST049.jpg inception v3.txt
INFO:tensorflow:Creating bottleneck at /tmp/bottleneck/a_rock/IMG_20191220_191651_BURST009.jpg inception v3.txt
INFO:tensorflow:Creating bottleneck at /tmp/bottleneck/a_rock/IMG_20191220_191651_BURST037.jpg inception v3.txt
INFO:tensorflow:Creating bottleneck at /tmp/bottleneck/a_rock/IMG_20191220_191720_BURST031.jpg inception v3.txt
INFO:tensorflow:Creating bottleneck at /tmp/bottleneck/a_rock/IMG_20191220_191730_BURST023.jpg inception v3.txt
INFO:tensorflow:Creating bottleneck at /tmp/bottleneck/a_rock/IMG_20191220_191730_BURST030.jpg inception v3.txt
```

Рисунок 11 – Создание текстовых файлов bottleneck

После создания текстовых файлов bottleneck начинается процесс обучения (рисунок 12).

```
INFO:tensorflow:2019-12-21 00:10:54.120104: Step 0: Cross entropy = 1.507861
INFO:tensorflow:2019-12-21 00:10:54.530664: Step 0: Validation accuracy = 25.0% (N=100)
INFO:tensorflow:2019-12-21 00:10:57.930546: Step 10: Train accuracy = 91.0%
INFO:tensorflow:2019-12-21 00:10:57.930919: Step 10: Cross entropy = 0.935152
INFO:tensorflow:2019-12-21 00:10:58.282476: Step 10: Validation accuracy = 82.0% (N=100)
INFO:tensorflow:2019-12-21 00:11:02.009554: Step 20: Train accuracy = 97.0%
INFO:tensorflow:2019-12-21 00:11:02.009881: Step 20: Cross entropy = 0.647566
INFO:tensorflow:2019-12-21 00:11:02.353814: Step 20: Validation accuracy = 95.0% (N=100)
INFO:tensorflow:2019-12-21 00:11:05.474315: Step 30: Train accuracy = 95.0%
INFO:tensorflow:2019-12-21 00:11:05.474621: Step 30: Cross entropy = 0.522539
INFO:tensorflow:2019-12-21 00:11:05.823989: Step 30: Validation accuracy = 91.0% (N=100)
INFO:tensorflow:2019-12-21 00:11:08.915052: Step 40: Train accuracy = 97.0%
INFO:tensorflow:2019-12-21 00:11:08.915362: Step 40: Cross entropy = 0.450809
INFO:tensorflow:2019-12-21 00:11:09.253221: Step 40: Validation accuracy = 95.0% (N=100)
INFO:tensorflow:2019-12-21 00:11:12.340524: Step 50: Train accuracy = 97.0%
INFO:tensorflow:2019-12-21 00:11:12.340849: Step 50: Cross entropy = 0.414332
INFO:tensorflow:2019-12-21 00:11:12.670920: Step 50: Validation accuracy = 95.0% (N=100)
INFO:tensorflow:2019-12-21 00:11:15.746956: Step 60: Train accuracy = 99.0%
INFO:tensorflow:2019-12-21 00:11:15.747266: Step 60: Cross entropy = 0.323118
INFO:tensorflow:2019-12-21 00:11:16.084330: Step 60: Validation accuracy = 96.0% (N=100)
INFO:tensorflow:2019-12-21 00:11:19.163789: Step 70: Train accuracy = 98.0%
INFO:tensorflow:2019-12-21 00:11:19.164207: Step 70: Cross entropy = 0.335107
INFO:tensorflow:2019-12-21 00:11:19.500304: Step 70: Validation accuracy = 99.0% (N=100)
INFO:tensorflow:2019-12-21 00:11:22.579544: Step 80: Train accuracy = 97.0%
INFO:tensorflow:2019-12-21 00:11:22.579925: Step 80: Cross entropy = 0.256131
INFO:tensorflow:2019-12-21 00:11:22.915284: Step 80: Validation accuracy = 97.0% (N=100)
INFO:tensorflow:2019-12-21 00:11:25.975275: Step 90: Train accuracy = 98.0%
INFO:tensorflow:2019-12-21 00:11:25.975641: Step 90: Cross entropy = 0.255813
INFO:tensorflow:2019-12-21 00:11:26.313926: Step 90: Validation accuracy = 98.0% (N=100)
INFO:tensorflow:2019-12-21 00:11:29.385284: Step 100: Train accuracy = 97.0%
INFO:tensorflow:2019-12-21 00:11:29.385676: Step 100: Cross entropy = 0.227215
INFO:tensorflow:2019-12-21 00:11:29.720799: Step 100: Validation accuracy = 99.0% (N=100)
INFO:tensorflow:2019-12-21 00:11:32.809113: Step 110: Train accuracy = 99.0%
INFO:tensorflow:2019-12-21 00:11:32.809482: Step 110: Cross entropy = 0.191436
INFO:tensorflow:2019-12-21 00:11:33.143289: Step 110: Validation accuracy = 98.0% (N=100)
```

Рисунок 12 – Процесс обучения нейронной сети

В зависимости от мощности оборудования спустя определённый промежуток времени нейронная сеть закончит обучение. Обученную нейронную сеть можно использовать для распознавания изображений на наличие дефектов.

2.8 Моделирование и анализ полученных результатов

На основе обученной ранее нейронной сети было произведено моделирование для распознавания различных дефектов.

Взято тестовое изображение камня, которое изображено на рисунке 13.



Рисунок 13 – Тестовое изображение камня

Для того, чтобы нейронная сеть смогла распознать, что изображено на картинке, тестовое изображение помещено в папку `defects_foto` и в терминале прописана команда: `python scripts/label_image.py`. На рисунке 14 показан результат моделирования.

```
Evaluation time (1-image): 2.207s  
a rock (score=0.98558)  
hole (score=0.00810)  
garbage (score=0.00600)  
purely (score=0.00028)  
joint (score=0.00004)
```

Рисунок 14 – Результат моделирования

Из результата моделирования видно, что нейронная сеть на 98% распознала на изображении камень, на 0,8% распознала отверстие, на 0,6% распознала мусор, на 0,02% распознала стык и только на 0,004% распознала, что в трубе нет никаких дефектов.

Далее взято следующее тестовое изображение, на которой изображено отверстие в трубе (рисунок 15).



Рисунок 15 – Тестовое изображение отверстия

На рисунке 16 показаны результаты моделирования.

```
Evaluation time (1-image): 2.881s
hole (score=0.99249)
joint (score=0.00458)
garbage (score=0.00148)
a rock (score=0.00140)
purely (score=0.00005)
```

Рисунок 16 – Результат моделирования

Из результата моделирования видно, что нейронная сеть на 99% распознала на изображении отверстие, на 0,4% распознала стык, на 0,1% распознала мусор, на 0,01% распознала камень и только на 0,005% распознала, что в трубе нет никаких дефектов.

Последующие результаты моделирования были сведены в таблицу 4.

Таблица 4 – Результаты моделирования

Номер моделирования	a_rock, %	hole, %	purely, %	garbage, %	joint, %
№1	98,1	0,55	0,022	1,24	0,003
№2	99,6	0,056	0,002	0,19	0,055
№3	0,029	0,029	0,001	99,9	0,001
№4	0,001	0,001	0	99,9	0
№5	0,35	98,5	0,006	0,74	0,35
№6	0,65	99	0,02	0,1	0,13
№7	0,03	0,1	0,04	0,04	99,7
№8	0,04	0,5	0,007	0,03	99,4
№9	0,09	0,009	99,6	0,2	0,008
№10	0,9	10,3	78,3	4,7	5,5

Из таблицы 4 видно, что обученная нейронная сеть с задачей справляется. В моделировании №1 и №2 были использованы изображения камня и нейросеть распознала верно в среднем на 98,85%. Также в моделировании №3 и №4 верно распознала мусор на 99,9%, в моделировании

№5 и №6 использовалось изображение отверстия, которое распознана в среднем на 98,75. В моделировании №7 и №8 верно распознан стык на 99,55%. В случае с моделированием №9 и №10, нейросеть распознала в среднем на 88,95%, что на изображении нет никаких дефектов и на 10,3%, что на изображении есть стык. Это связано с тем, что изображения стыка и изображения, где нет никаких дефектов, достаточно похожи, что бы нейронная сеть смогла распознать их ближе к 99%.

2.9 Выводы

Разработан функциональный состав устройства управления змеевидным роботом и определены основные принципы функционирования систем. На основе разработанной функциональной схемы разработаны базовые алгоритмы функционирования УУ робота, что позволило осуществить разработку интеллектуальной системы управления. Определено, что в данном классе задач, а именно для распознавания изображений наибольший эффект следует ожидать от сверточной нейронной сети.

Выполнен обзор средств моделирования и определён инструментарий для создания модели нейронной сети. На основе обзора была выбрана библиотека TensorFlow.

С помощью библиотеки TensorFlow и языка программирования Python разработано алгоритмическое и программное обеспечение УУ, создана программная модель.

В лабораторных условиях изготовлена модель трубы, определен ряд дефектов для лабораторного демонстратора и получены тестовые изображения для обучения нейронной сети. Нейронная сеть прошла обучение.

Выполнена отработка полученных решений и программное моделирование, которое показало, что обученная нейронная сеть верно распознает дефекты в трубе.

Таким образом предполагаемая научная новизна исследований заключается в способе интеллектуального, автономного управления поисковым роботом, в основу которого положен принцип принятия решений о пространственном перемещении, с параллельным детектированием аномалий по данным видеонаблюдения, получаемым в режиме реального времени.

Полученные результаты позволили перейти к выбору компонентов и сборке лабораторного макета змеевидного робота.

3 Разработка лабораторного макета змеевидного робота

3.1 Выбор компонентов для разработки лабораторного макета змеевидного робота

На основании функциональной схемы устройства управления змеевидного робота (Глава 2), был произведен выбор составляющих компонентов для лабораторного макета змеевидного робота.

3.1.1 Выбор микропроцессорного ядра

Определено, что главным компонентом системы является вычислительное ядро, реализуемое на основе промышленного одноплатного вычислителя, функционирующего под управлением микроконтроллера.

Выделены 2 категории плат: платы на микроконтроллере и одноплатные компьютеры. Основное отличие данных решений – это наличие или отсутствие режима многозадачности и возможность функционирования под управлением ОС. Выявлено, что одноплатные компьютеры исполняют программы в рамках операционной системы (чаще всего Linux), обладают большей производительностью и широкими мультимедийными возможностями [22]. Результаты сравнения выбранных решений представлены в таблице 5 [23].

Таблица 5 – Сравнительная таблица микроконтроллеров и одноплатных компьютеров

Характеристика	Микроконтроллер	Одноплатный компьютер
Производительность	1 ядро, десятки-сотни МГц, десятки КБ оперативки, десятки-сотни КБ постоянной памяти	1 или более ядер, сотни-тысячи МГц, сотни МБ оперативки, гигабайты постоянной памяти

Окончание таблицы 5

Характеристика	Микроконтроллер	Одноплатный компьютер
Многозадачность	Нет, но можно эмулировать	Да, управляется ОС
Интернет	Обычно нужны дополнительные модули и глубокое знание протоколов	Легко подключается из коробки, сетевой модуль обычно уже на борту
Работа от батареек	Потребляет единицы-десятки мА. Возможны недели работы от батареек	Потребляет сотни-тысячи мА. Заряда большого аккумулятора хватит от силы на десяток часов
Языки программирования	Ограниченный. Чаще C/C++	Python, JavaScript, Bash и десятки других: любые доступные в ОС
Работа с видео, компьютерное зрение	Не хватит мощности	OpenCV, аппаратные видекодеки, HDMI-выход

Для решения поставленных задач выбран одноплатный компьютер. Он позволяет не только работать с видео, но и использовать библиотеку OpenCV (библиотека компьютерного зрения с открытым исходным кодом). Кроме имеется возможность установки операционной системы и использования языков программирования, например, Python, который хорошо зарекомендовал себя в области машинного обучения.

В качестве одноплатного компьютера был выбран Raspberry Pi 3 model B – одноплатный компьютер с поддержкой беспроводной локальной сети и Bluetooth (Рисунок 17).



Рисунок 17 – Одноплатный компьютер Raspberry Pi 3 model B

Данный одноплатный компьютер имеет следующие характеристики:

- 4-хъядерный процессор 1,2 ГГц Broadcom BCM2837 64-битный процессор;
- 1 ГБ ОЗУ;
- Беспроводная локальная сеть BCM43438 и Bluetooth Low Energy (BLE);
- 100 базовых сетей Ethernet;
- 40-контактный удлиненный GPIO;
- 4 порта USB 2;
- 4-полюсный стереовыход и композитный видеопорт;
- Полноразмерный HDMI;
- Порт камеры CSI для подключения камеры Raspberry Pi;
- Порт дисплея DSI для подключения сенсорного дисплея Raspberry Pi;
- Порт Micro SD для загрузки операционной системы и хранения данных;
- Источник питания Micro USB до 2,5 А [24].

3.1.2 Выбор видеокamеры для одноплатного компьютера

Анализ рынка позволил определить модель видеокamеры - RPi Camera (B) – это камера с регулируемым фокусом для одноплатного компьютера Raspberry Pi (Рисунок 18). Камера совместима со всеми версиями миникомпьютера.



Рисунок 18 – Камера RPi Camera (B)

Камера RPi Camera (B) имеет следующие характеристики:

- Сенсор 5 megapixel OV5647;
- Размер CCD: 1/4-дюйма;
- Диафрагма (F): 2.0;
- Фокусное расстояние: 6мм (регулируемое);
- Диагональ: 75.7 градусов;
- Лучшее разрешение сенсора: 1080p;
- Размеры: 32 x 32мм [25].

3.1.3 Выбор датчика препятствий

Определено, что для контроля наличия препятствий на пути робота необходимо как минимум два специальных датчика. Цифровой инфракрасный датчик препятствий YL-63 (рисунок 19) применяется тогда, когда нужно определить наличие объекта, а точное расстояние до объекта знать необязательно. Датчик состоит из инфракрасного излучателя и фотоприемника. Он построен на основе компаратора LM393, который выдает напряжение на выход по принципу: обнаружено препятствие – логический уровень HIGH, не обнаружено – логический уровень LOW, данное состояние показывает и находящийся на датчике красный светодиод. Пороговое значение зависит от настройки датчика и регулируется с помощью установленного на модуле потенциометра.

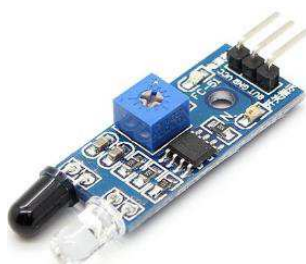


Рисунок 19 – Инфракрасный датчик препятствий YL-63

Ниже приведены технические характеристики инфракрасного датчика препятствий YL-63:

- Напряжение питания: 3.3–5 В;
- Компаратор: LM393;
- Расстояние обнаружения препятствий: 2 – 30 см;
- Эффективный угол обнаружения препятствий: 35°;
- Потенциометр для изменения чувствительности;
- Светодиод индикации питания;
- Светодиод индикации срабатывания;
- Размеры: 43 x 16 x 7 мм [26].

3.1.4 Выбор мотора и колес для передвижения робота

Мотор-редуктор – мотор для создания подвижных платформ (роботов, радиоуправляемых моделей) (Рисунок 20). Хорошо совместим с колесами для роботов (Рисунок 21).

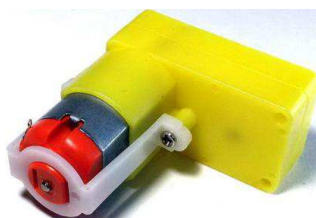


Рисунок 20 – Мотор-редуктор 90°

Технические характеристики мотора-редуктора приведены ниже:

- Рабочее напряжение: 3-12 В;
- Номинальное напряжение: 6 В;
- Крутящий момент: 0.8 кг/см (3 В);
- Передаточное число: 1:48;

- Размер: 53x47x22 мм;
- Вес: 30 г [27].



Рисунок 21 – Колесо для робота

3.1.5 Выбор платформы для крепления компонентов

Для подключения выбранных компонентов необходимо выбрать специальную платформу, которая позволяет закрепить и расположить все элементы. Для такой цели подходит платформа AlphaBot – это платформа для создания мобильного робота совместимая с Raspberry Pi и Arduino (Рисунок 22).

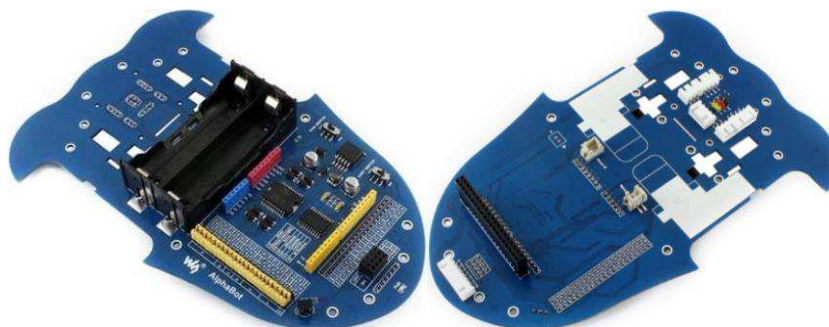


Рисунок 22 – Платформа AlphaBot

Особенности данной конструкции заключаются в следующем:

- Raspberry Pi/Arduino интерфейсы, может работать сразу с двумя платформами;
- Arduino разъем расширения, поддерживает Shields;
- Модульная конструкция, простое подключение датчиков;
- LM298P драйвер двигателей с диодной схемой защиты, для большей безопасности;
- LM2596 регулятор напряжения, обеспечивает стабильное напряжение 5В для Raspberry Pi и Arduino;
- TLC1543 АЦП, позволяет Raspberry Pi работать с аналоговыми датчиками [28].

Для работы данной платформы необходимо два аккумулятора Li-ion, 2400 mAh, 3.7V (Рисунок 23), которые предназначены для использования в фонарях и других устройствах с высоким энергопотреблением.



Рисунок 23 – Аккумуляторы

3.1.6 Выбора сервоприводов для поворота камеры

Повороты камеры будет осуществлять пара сервоприводов TowerPro SG90 – электромеханический прибор, состоящий из мотора постоянного тока, редуктора и управляющей электроники. Выходной вал поворачивается на

задаваемый угол 180° – по 90° в каждом направлении. На рисунке 24 изображен внешний вид сервопривода TowerPro SG90.



Рисунок 24 – Сервопривод TowerPro SG90

Сервопривод Tower Pro SG90 применяется для поворота деталей различных механизмов. Благодаря редуктору на выходном валу развивается усилие достаточное для применения в подвижных роботах. Низкая скорость поворота вала позволяет более точно получить требуемый угол поворота.

Технические характеристики сервопривода Tower Pro SG90:

- Напряжение питания: 4,5 – 5,5В
- Максимальный ток: 1А
- Крутящий момент: 1,8 килограмм/сантиметр
- Температура окружающего воздуха при работе: 0 – 55 °С
- Длина жгута: 230 мм [29].

3.2 Сборка лабораторного макета

В лабораторных условиях был собран макет ведущей секции робота из ранее выбранных компонентов (Рисунок 25).

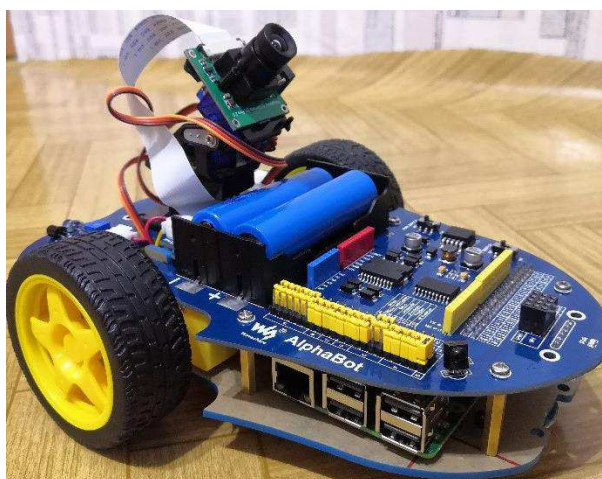


Рисунок 25 – Макет ведущей секции змеевидного робота

Собранный макет состоит из следующих компонентов:

- Одноплатный компьютер Raspberry Pi 3 model B – позволяет управлять всеми составляющими системы;
- Видеокамера RPi Camera (V) – необходима для съемки фотографий при передвижении робота;
- Два сервопривода, которые осуществляют поворот камеры как по горизонтали, так и по вертикали;
- Два цифрового инфракрасного датчика препятствий YL-63 – служит для контроля наличия препятствия на пути робота;
- Два мотора-редуктора на которых закреплены по два колеса необходимы для осуществления передвижения робота;
- Платформа AlphaBot – на данной платформе размещены и закреплены все составляющие компоненты робота;
- Две аккумуляторные батареи.

3.3 Выводы

Решена задача в выборе компонентов для лабораторного макета змеевидного робота. Так в состав макета ведущей секции робота входят:

- Одноплатный компьютер Raspberry Pi 3 model B;
- Видеокамера RPi Camera (V);
- Два сервоприводов;
- Два цифровых инфракрасных датчиков препятствий YL-63;
- Два мотора-редуктора на которых закреплены по два колеса;
- Платформа AlphaBot;
- Две аккумуляторные батареи

Для каждого компонента из состава макета разработаны тестовые программы и драйвера нижнего уровня, а также выполнена отработка сопряжения устройств с МЯ. Это позволило перейти к разработке алгоритмического и программного обеспечения макета верхнего уровня.

4 Разработка алгоритмического и программного обеспечения для лабораторного макета

4.1 Разработка алгоритмов управления

Перед разработкой программного обеспечения для лабораторного макета составлен алгоритм в виде блок-схем. Общий алгоритм функционирования робота представляет следующее:

1. Движение робота осуществляется посредством двух моторов-редукторов.
2. Далее видеочамера делает снимок полости трубопровода.
3. Сделанный снимок передается в нейронную сеть для определения наличия дефектов
4. Полученный результат обрабатывается и результаты записываются в память устройства.

На рисунке 26 представлен общий алгоритм движения робота в режиме объезда препятствий.

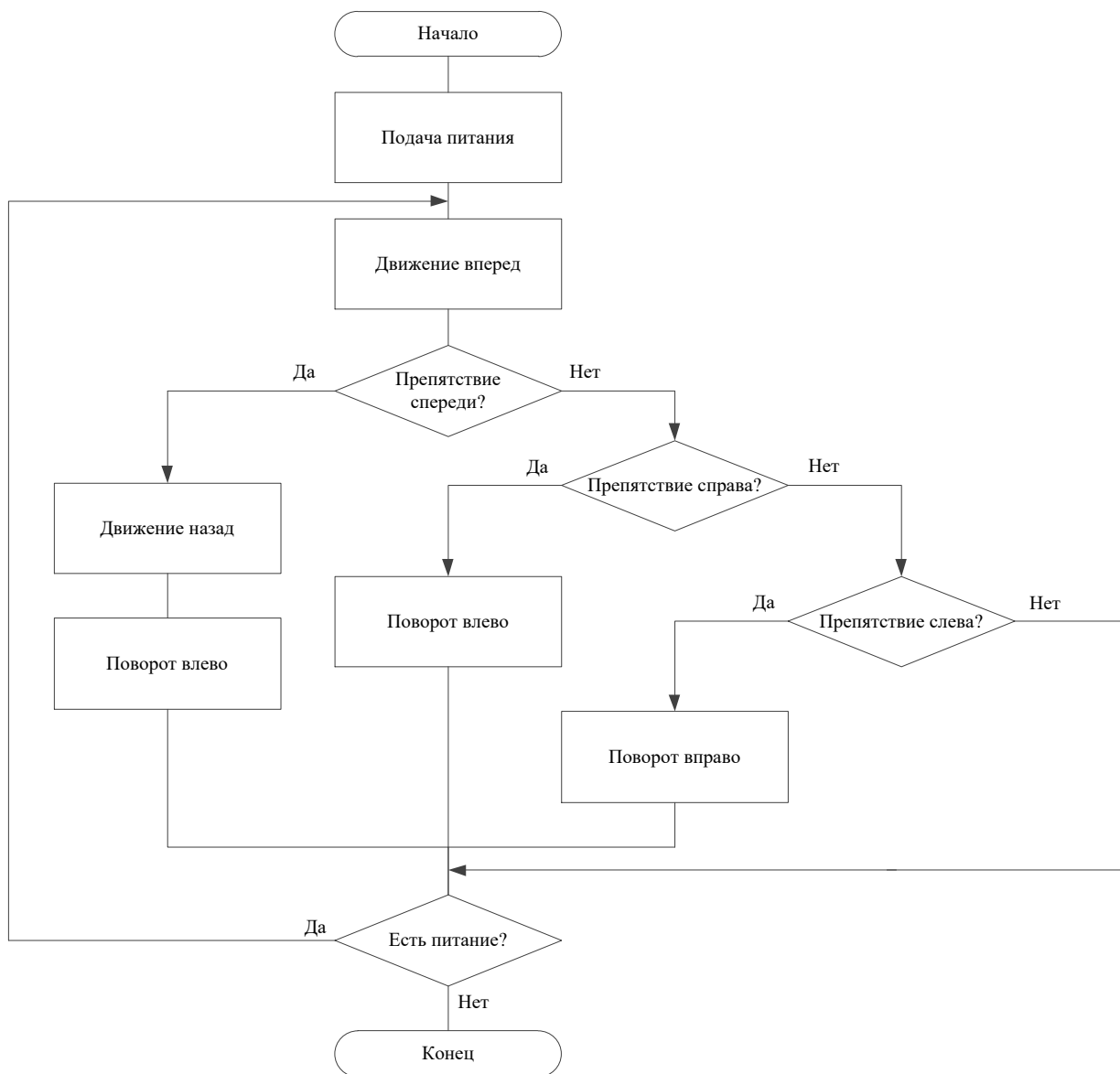


Рисунок 26 – Блок-схема движения робота

На рисунке 27 представлен общий алгоритм функционирования УУ в режиме обнаружения дефектов.

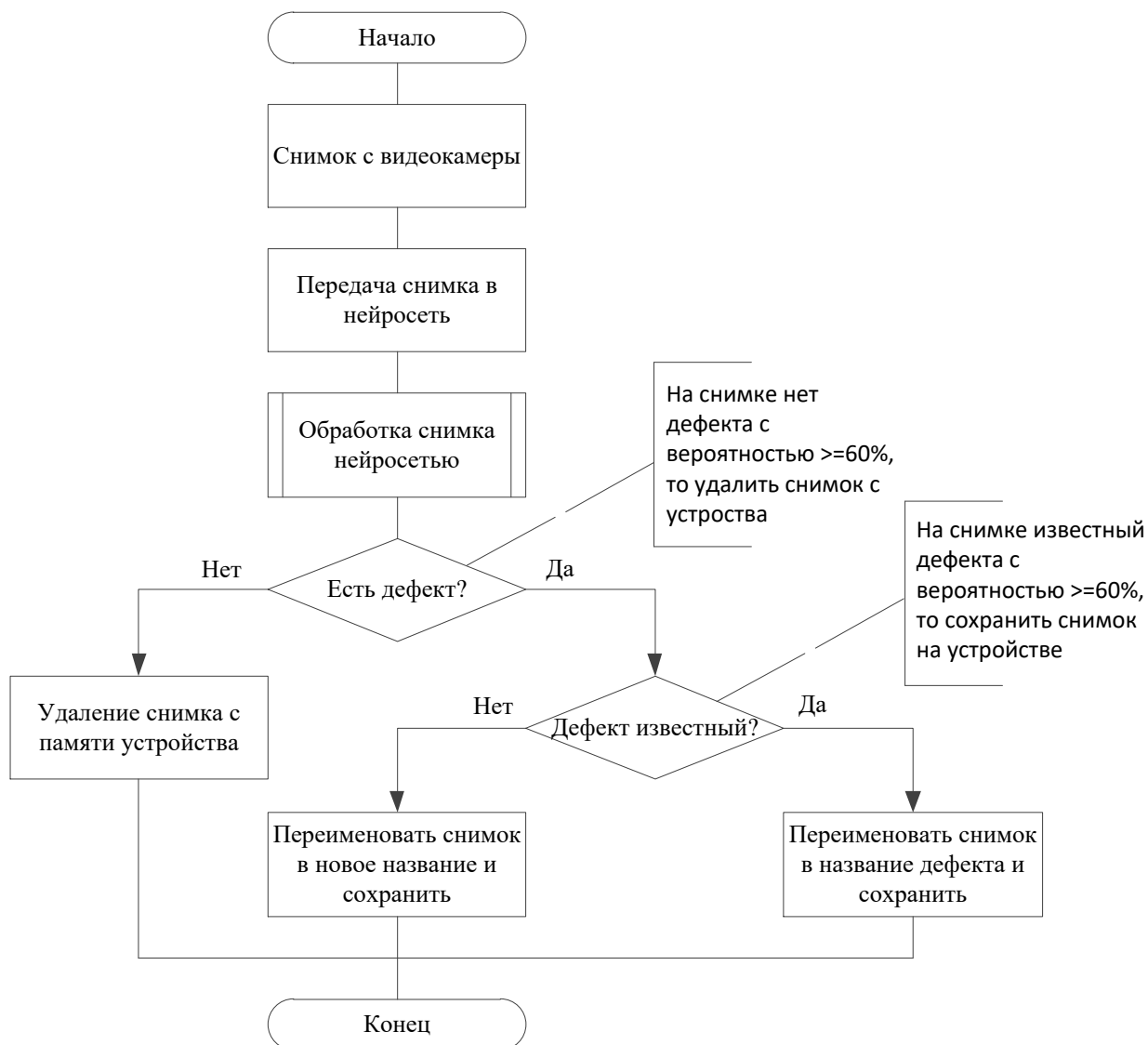


Рисунок 27 – Блок-схема обнаружения дефектов

4.2 Разработка программного обеспечения

Реализация разработанных алгоритмов УУ, в виде встроенного ПО для лабораторного макет работа, выполнено в операционной системе Raspbian с использованием языка программирования Python и библиотеки TensorFlow.

4.2.1 Разработка программного обеспечения движения робота

Ниже (Листинг 1) представлен основной фрагмент кода, где осуществляется передвижение робота с объездом препятствий (более подробно см. приложение В):

```
DR_status = GPIO.input(DR) #Считывание значения с правого датчика
препятствий
DL_status = GPIO.input(DL) #Считывание значения с левого датчика
препятствий
if((DL_status == 1) and (DR_status == 1)): #Если нет препятствий
    Ab.forward()
elif((DL_status == 1) and (DR_status == 0)): #Если препятствие справа
    Ab.left()
elif((DL_status == 0) and (DR_status == 1)): #Если препятствие слева
    Ab.right()
else: #Иначе - препятствие спереди
    Ab.backward()
    time.sleep(0.2)
    Ab.left()
    time.sleep(0.2)
```

Листинг 1 – Код ПО – объезд препятствий

С помощью команд `DR_status = GPIO.input(DR)` и `DL_status = GPIO.input(DL)` происходит считывание статусов правого и левого датчика препятствий в данный момент времени.

Далее на основе считанных значений происходит их проверка. Если `DL_status = 1` и `DR_status = 1`, то движение робота осуществляется вперед с помощью команды `Ab.forward()`, если `DL_status = 1` и `DR_status = 0`, то

осуществляется поворот влево с помощью команды `Ab.left()`, если `DL_status = 0` и `DR_status = 1`, то движение робота осуществляется вправо с помощью команды `Ab.right()`, иначе, если `DL_status = 0` и `DR_status = 0`, то движение робота осуществляется назад и влево с помощью команд `Ab.backward()` и `Ab.left()` соответственно.

4.2.2 Разработка программного обеспечения обнаружения дефектов

Ниже (Листинг 2) представлен основной фрагмент кода, где осуществляется захват изображения и передачи снимка в нейронную сеть (более подробно см. приложение В):

```
i = 0
time.sleep(2)
camera.capture('/home/pi/Desktop/diplom/foto/' + str(i) + '.jpg')
ns = 'python3 scripts/label_image.py --image ' + str(i) + '.jpg'
os.system(ns)
i++
```

Листинг 2 – Код ПО – Захват и передача изображения в нейронную сеть

Для настройки фокуса камеры перед осуществлением снимка используется пауза в две секунды с помощью команды `time.sleep(2)`. Далее осуществляется захват изображения и сохранения изображения по указанному пути с использованием команды `camera.capture('/home/pi/Desktop/diplom/foto/' + str(i) + '.jpg')`. Команды `ns = 'python3 scripts/label_image.py --image ' + str(i) + '.jpg'` и `os.system(ns)` передают изображение в нейронную сеть для обработки.

После обработки нейронной сетью изображения, идет обработка результата. Основной код обработки результата представлен ниже (Листинг 3):

```

for i in top_k: #Обход вариантов дефектов
    if labels[i] == 'purely' and results[i] >= 0.60: #Если нет дефекта
        remove = 'rm ' + file_name
        os.system(remove) #Удалить снимок
    elif results[i] >= 0.60: #Если есть дефект
        rename = 'mv ' + file_name + ' ' + labels[i] +
            dt.datetime.now().strftime('%d-%m-%Y-%H-%M-%S') + '.jpg'
        os.system(rename) #Переименовать снимок
    else #Иначе - сохранить снимок (что-то новое на снимке)
        rename = 'mv ' + file_name + ' ' + 'new_defect' +
            dt.datetime.now().strftime('%d-%m-%Y-%H-%M-%S') +
            '.jpg'
        os.system(rename) #Сохранить снимок

```

Листинг 3 – Код ПО – Обработка результата

С помощью цикла `for i in top_k` осуществляется обход всех вариантов дефектов. На каждой итерации цикла идет сравнение результата. Условие `if labels[i] == 'purely' and results[i] >= 0.60` осуществляет проверку наличия дефекта на снимке. Если оно истинно – снимок удаляется с памяти устройства, что означает отсутствие дефекта, иначе идет проверка условия `elif results[i] >= 0.60`. Если оно истинно – снимок сохраняется в память устройства, что означает наличие дефекта, в качестве имени файла указывается название дефекта и дата создания снимка `rename = 'mv ' + file_name + ' ' + labels[i] + dt.datetime.now().strftime('%d-%m-%Y-%H-%M-%S') + '.jpg'`. Если же данное условие ложно – снимок сохраняется с новым именем дефекта `rename = 'mv ' + file_name + ' ' + 'new_defect' + dt.datetime.now().strftime('%d-%m-%Y-%H-%M-%S') + '.jpg'`. Это значит, что нейронная сеть не смогла распознать на снимке ни один из дефектов, и был найден новый дефект.

4.3 Проведение экспериментов

Для проверки работоспособности собранного макета змеевидного робота, а также проверки разработанного программного обеспечения, был проведен ряд экспериментов, позволяющий наглядно увидеть, справляется ли змеевидный робот и его УУ со своей задачей. Для демонстрации работы по каждому виду дефектов было сделано по 40 фотографий. Затем была обучена нейронная сеть. Процесс создания снимков и обучения нейронной сети был идентичен описанному в главе 2.

После обучения нейронной сети было определено пороговое значение, при котором нейронная сеть будет верно распознавать дефект на снимке. Для этого было произведено 8 экспериментов, в каждом из которых нейронная сеть распознавала различные объекты на снимке отличные от известных ей дефектов. В каждом из экспериментов показано процентное отношение к одному из видов дефектов: наличие мусора, наличия отверстия, наличие камня, наличие стыка или же отсутствие дефектов в трубе. Результаты экспериментов приведены в таблице 5.

Таблица 5 – Результаты проведения экспериментов

Вид дефекта	Эксперимент							
	Э1	Э2	Э3	Э4	Э5	Э6	Э7	Э8
Мусор, %	37	26	16	48	0,03	6	14	54
Отверстие, %	24	27	46	23	56	25	54	8
Камень, %	29	24	20	10	10	8	7	21
Стык, %	7	16	15	14	5	2	6	6
Нет дефектов, %	0,8	5	1	3	28	58	17	9

В столбцах с именами Э1...Э8 обозначены эксперименты со следующими объектами на снимках:

- Э1 – Снимок с изображением принтера;
- Э2 – Снимок с изображением ноутбука;
- Э3 – Снимок с изображением телефона;

- Э4 – Снимок с изображением ручки;
- Э5 – Снимок с изображением батарейки в трубе;
- Э6 – Снимок с изображением точилки в трубе;
- Э7 – Снимок с изображением флешка в трубе;
- Э8 – Снимок с изображением скотча в трубе.

Исходя их эксперимента Э6, где на снимке был изображен объект в виде точилки в трубе, то нейронная сеть на 58% распознала, что в трубе нет никаких дефектов, что является неверным. Это самый максимальный показатель, который был достигнут в ходе проведения экспериментов. Таким образом, можно сделать вывод, что пороговое значение, после которого нейронная сеть будет верно распознавать дефекты, является значение 58%.

4.3.1 Эксперимент с дефектом в виде наличия камня в трубе

На рисунке 28 изображен макет змеевидно робота, который расположен внутри макета трубы, в которой помещен камень.

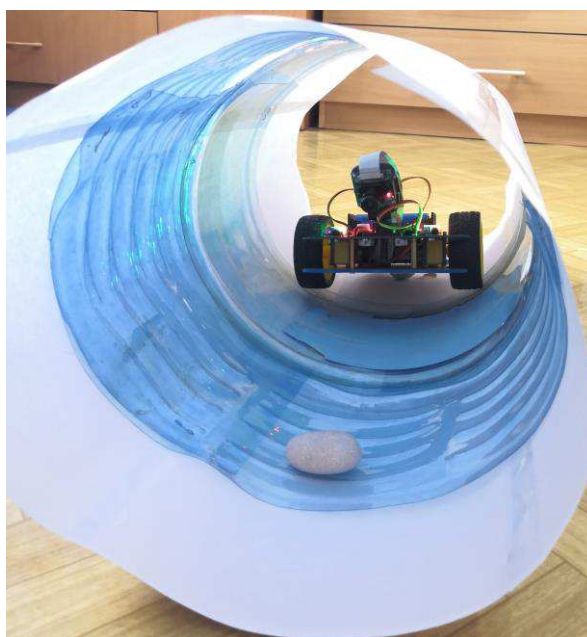


Рисунок 28 – Эксперимент с дефектом в виде наличия камня в трубе

Далее видеокамера делает снимок внутренности трубы. Изображение, получившееся в результате снимка, изображено на рисунке 29.



Рисунок 29 – Снимок, сделанный видеокамерой робота

Снимок передается в нейронную сеть и обрабатывается. Результаты представлены на рисунке 30.

```
Evaluation time (1-image): 4.366s  
rock (score=0.62601)  
garbage (score=0.22212)  
purely (score=0.07237)  
hole (score=0.06511)  
joint (score=0.01439)
```

Рисунок 30 – Результаты обработки изображения нейронной сетью

Результаты показывают, что нейронная сеть на 62% распознала на изображении дефект в виде камня, что является верным. На 22% распознано мусор, т.к. изображения камня и мусора достаточно похожи. И ниже 10% распознано все остальные дефекты.

4.3.2 Эксперимент с дефектом в виде наличия отверстия в трубе

На рисунке 31 изображен макет змеевидно робота, который расположен внутри макета трубы, в которой проделано отверстие.



Рисунок 31 – Эксперимент с дефектом в виде наличия отверстия в трубе

Далее видеокамера делает снимок внутренности трубы. Изображение, получившееся в результате снимка, изображено на рисунке 32.



Рисунок 32 – Снимок, сделанный видеокамерой робота

Снимок передается в нейронную сеть и обрабатывается. Результаты представлены на рисунке 33.

```
Evaluation time (1-image): 4.318s  
hole (score=0.82587)  
purely (score=0.14653)  
joint (score=0.01675)  
rock (score=0.00709)  
garbage (score=0.00377)
```

Рисунок 33 – Результаты обработки изображения нейронной сетью

Результаты показывают, что нейронная сеть на 82% распознала на изображении дефект в виде отверстия в трубе, что является верным. И только на 14% распознала, что в трубе нет никаких дефектов. И ниже 2% распознала все остальные дефекты.

4.3.3 Эксперимент с отсутствием дефектов

На рисунке 34 изображен макет змеевидно робота, который расположен внутри макета трубы, в которой отсутствуют какие-либо дефекты.

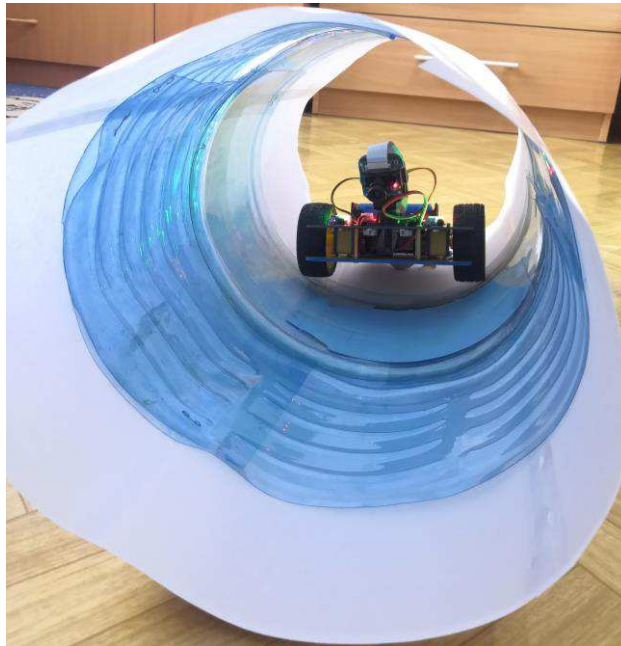


Рисунок 34 – Эксперимент с отсутствием дефектов в трубе

Далее видеочамера делает снимок внутренности трубы. Изображение, получившееся в результате снимка, изображено на рисунке 35.

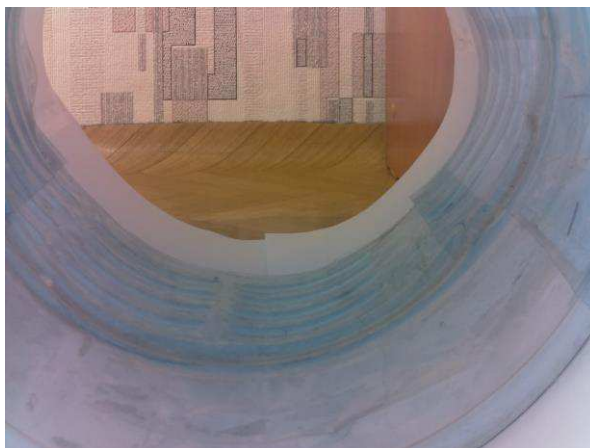


Рисунок 35 – Снимок, сделанный видеочамерой робота

Снимок передается в нейронную сеть и обрабатывается. Результаты представлены на рисунке 36.

```
Evaluation time (1-image): 4.391s
purely (score=0.76296)
hole (score=0.17957)
joint (score=0.03690)
rock (score=0.01422)
garbage (score=0.00635)
```

Рисунок 36 – Результаты обработки изображения нейронной сетью

Результаты показывают, что нейронная сеть на 76% распознала, что внутри трубы отсутствуют какие-либо дефекты, что является верным. И только на 17% распознала, что в трубе присутствует дефект, в виде отверстия. И ниже 4% распознала все остальные дефекты.

4.4 Выводы

Разработано алгоритмическое и программное обеспечение УУ змеевидного робота. В том числе разработаны модули передвижения робота, а также модули получения снимков с видеокамеры и модули управления нейронной сетью. Разработанное ПО установлено на лабораторный демонстратор и выполнена его отработка в тестовом режиме. Выявленные ошибки ПО устранены. Макет ведущей секции подготовлен к экспериментальной отработке в составе лабораторного демонстратора.

На собранном лабораторном демонстраторе проведены эксперименты, которые показали, что пороговое значение результатов обучения, после которого нейронная сеть будет верно распознавать дефекты на снимках составляет 58%.

Для демонстрации корректности кода, а также правильности обучения нейронной сети, был проведен ряд экспериментов. При этом:

Эксперимент с наличием камня в трубе показал, что нейронная сеть верно распознала дефект в трубе на 62%, что выше порогового значения.

Эксперимент с наличием отверстия в трубе показал, что нейронная сеть верно распознала дефект в трубе на 82%, что выше порогового значения.

Эксперимент с отсутствием дефектов в трубе показал, что нейронная сеть верно распознала отсутствие дефектов в трубе 76%, что выше порогового значения.

Таким образом обученная нейронная сеть с поставленной задачей справляется.

Эксперименты с движением колесной секции показали адекватную реакцию робота на наличие и/или отсутствие препятствий, что подтверждает правильность функционирования этой секции ПО.

Таким образом, посредством лабораторного демонстратора показан факт достижения цели ВКР и решения всех поставленных задач.

ЗАКЛЮЧЕНИЕ

На основании анализа известных решений была сформулирована цель ВКР и поставлены задачи для ее достижения. Рассмотрев доступные аналоги определено, что роботы, обладающие достаточно высокими массогабаритными характеристиками и связанные с объектом управления посредством кабеля, являются наиболее распространёнными. Автономные роботы практически не применяются. Выявлена необходимость получать более лучшие результаты в инспекции трубопроводов. Определено, что этого можно достичь путем разработки новых интеллектуальных методик и технологий исследования технологических трубопроводов. Выявлено, что среди известных подходов наибольший интерес вызывает технологии видеодиагностики, поскольку такой подход позволяет выявить максимум возможных дефектов.

Разработанный функциональный состав устройства управления змеевидным роботом позволил осуществить разработку интеллектуальной системы управления. Выявлено, что для распознавания изображений наибольший эффект следует ожидать от сверточной нейронной сети. На основе обзора средств моделирования, выбранная библиотека TensorFlow с использованием языка программирования Python, подходит для разработки алгоритмического и программного обеспечения УУ. В лабораторных условиях собрана модель трубы и определён ряд дефектов. По каждому виду дефектов сделано несколько изображений для обучения нейронной сети. Результаты моделирования показали, что обученная нейронная сеть верно распознает дефекты в трубе.

Из выбранных компонентов в лабораторных условиях собран макет змеевидного робота, состоящий из: одноплатный компьютер Raspberry Pi 3 model B, видеочамера RPi Camera (B), два сервопривода, два цифровых

инфракрасных датчиков препятствий YL-63, два мотора-редуктора на которых закреплены по два колеса, платформа AlphaBot, две аккумуляторные батареи.

Для собранного макета разработано алгоритмическое и программное обеспечение. Эксперименты, что пороговое значение результатов обучения, после которого нейронная сеть будет верно распознавать дефекты на снимках составляет 58%. Для демонстрации корректности кода, а также правильности обучения нейронной сети, был проведен ряд экспериментов, который показал, что обученная нейронная сеть с поставленной задачей справляется. Эксперименты с движением колесной секции показали адекватную реакцию робота на наличие и/или отсутствие препятствий, что подтверждает правильность функционирования этой секции ПО.

Таким образом, посредством лабораторного демонстратора показан факт достижения цели ВКР и решения всех поставленных задач.

СПИСОК СОКРАЩЕНИЙ

ВКР – Выпускная квалификационная работа;

АЭС – Атомная электростанция;

ЗР – Змеевидный робот;

УУ – Устройство управления;

МЯ – Микропроцессорное ядро;

ИС – Интеллектуальная система;

ПО – Программное обеспечение.

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. Поезжаева Е. В., Федотов А. Г., Заглядов П. В. Разработка робота для контроля трубопроводов // Молодой ученый. — 2015. — №16. — С. 218-222. — URL <https://moluch.ru/archive/96/21567/>
2. ТАИНА-SVARKI – всё о сварке металлов. Магнитные методы контроля сварных швов. Магнитная дефектоскопия сварки. [Электронный ресурс] // URL: <https://taina-svarki.ru/kachestvo-i-kontrol-svarki/kontrol-svarnyh-soedineniy/magnitnye-metody-kontrol-svarnyh-shvov-magnitnaya-defektoskopiya.php>
3. Группа компаний «МЦК». Контроль трубопроводов, диагностика и ультразвуковой контроль сварных соединений трубопроводов. [Электронный ресурс] // URL: <https://www.mosexp.ru/docs/pipeline>
4. «Энергосвет» - всё об энергосбережении в интернете. Опыт опрессовки трубопроводов тепловых сетей на повышенное давление [Электронный ресурс] // URL: <http://www.energsovet.ru/nadegts.php?idd=32>
5. TrubaMaster.RU – Профессионально о трубах. Диагностика технологических трубопроводов: назначение и методы проведения работ. [Электронный ресурс] // URL: <http://trubamaster.ru/obslyuzhivanie/diagnostika-truboprovodov.html>
6. Santehclab.ru – Инженерные коммуникации. Видеодиагностика труб [Электронный ресурс] // URL: <http://santehclab.ru/uslugi/videodiagnostika-trub/>
7. RoboCraft. Ru – Роботы? Это просто! Змеелок–3 - робот-змея из Санкт-Петербурга [Электронный ресурс] // URL: <http://robocraft.ru/blog/news/3172.html>
8. Иванов А. А., Шмаков О. А., Демидов Д. А., Экспериментальное исследование змеевидного робота «ЗМЕЕЛОК 3» // Научно-технические ведомости СПбГПУ 1`2013. Информатика. Телекоммуникации. Управление. Санкт-Петербург, Россия URL:

<https://cyberleninka.ru/article/v/eksperimentalnoe-issledovanie-zmeevidnogo-robota-zmeelok-3>

9. Профоборудование. Оборудование для телеинспекций. Система телеинспекции Кроулер S300 (Великобритания) робот [Электронный ресурс] // URL: <https://teleinspekciya.ru/robotizirovannaja-sistema-teleinspekicii-krouler-s300-velikobritaniija>

10. Оборудование профессионалам – производство и поставки оборудования для чистки и диагностики трубопроводов по всей России и в страны СНГ. РОБОТ ДЛЯ ТЕЛЕИНСПЕКЦИИ ТРУБОПРОВОДОВ S-100 [Электронный ресурс] // URL: <https://xn---7sbbabgnkb1ad1bofffbbgcrnj0aa2b0b.xn--p1ai/teleinspekciija-truboprovodov/roboty/s-100>

11. Интеллектуальные системы в информационно аналитических (ситуационных) центрах в условиях перехода к цифровой экономике / Федотова М.А., Мартынова И.А. Московский экономический журнал №1 2019г. Стр. 4

12. Остроух А. В., Интеллектуальные системы/ А. В. Остроух // Красноярск: Научно-инновационный центр, 2015. – 110 с.

13. Hi-News.ru. – Новости о высоких технологиях. Нейронные сети [Электронный ресурс] // URL: <https://hi-news.ru/tag/nejronnye-seti>

14. Gradient-Based Learning Applied to Document Recognition / Yann LeCun, Leon Bottou, Yoshua Bengio, and Patrick Haffner. PROC. OF THE IEEE, NOVEMBER 1998, Стр. 10-20

15. Andreyex. ИТ Блог. Администрирование серверов на основе Linux (Ubuntu, Debian, CentOS, openSUSE). TensorFlow – сверточные нейронные сети [Электронные ресурс] // URL: <https://andreyex.ru/tensorflow-mashinnoe-obuchenie/tensorflow-osnovy-copy>

16. Neurohive – Нейронные сети. Рекуррентные нейронные сети: типы, обучение, примеры и применение [Электронный ресурс] // URL: <https://neurohive.io/ru/osnovy-data-science/rekurrentnye-nejronnye-seti/>
17. Amazon Web Services (AWS) [Электронный ресурс] // URL: <https://aws.amazon.com/ru/>
18. Teachable Machine. Train a computer to recognize your own images, sounds, & poses [Электронный ресурс] // URL: <https://teachablemachine.withgoogle.com/>
19. TensorFlow. An end-to-end open source machine learning platform [Электронный ресурс] // URL: <https://www.tensorflow.org/>
20. Computerworld. ИТ-индустрия – новости, обзоры, аналитика, продукты и услуги. Что такое TensorFlow и как это используется? [Электронный ресурс] // URL: <https://www.computerworld.ru/articles/Chto-takoe-TensorFlow-i-kak-eto-ispolzuetiya>
21. TensorFlow. An end-to-end open source machine learning platform [Электронный ресурс] // URL: <https://www.tensorflow.org/install?hl=ru>
22. Научно-популярный портал ЗАНИМАТЕЛЬНАЯ РОБОТОТЕХНИКА. Arduino или Raspberry Pi: какая платформа лучше? [Электронный ресурс] // URL: <http://edurobots.ru/2014/09/arduino-ili-raspberry-pi-kakaya-platforma-luchshe/>
23. Амперка – Всё для Arduino и Raspberry Pi. Arduino или Raspberry Pi: как выбрать контроллер для проекта [Электронный ресурс] // URL: <https://amperka.ru/page/development-board-guide>
24. Teach, Learn, and Make with Raspberry Pi – Raspberry Pi. Raspberry Pi 3 Model B [Электронный ресурс] // URL: <https://www.raspberrypi.org/products/raspberry-pi-3-model-b/>
25. ЧИП и ДИП - интернет-магазин приборов и электронных компонентов. RPi Camera (B), Камера для Raspberry Pi Model B+/2/3,

регулируемый фокус, угол обзора 72 гр [Электронный ресурс] // URL: <https://www.chipdip.ru/product/rpi-camera-b>

26. Запчасти и комплектующие для 3D принтера, Arduino, ЧПУ и робототехники в Москве | 3DIY. Инфракрасный датчик препятствий YL-63 [Электронный ресурс] // URL: <https://3d-diy.ru/wiki/arduino-datchiki/infrakrasnyj-datchik-prepyatstvij-yl-63/>

27. yourduino.ru - Электронные компоненты, Микроконтроллеры Arduino, Сенсоры и Радиодетали. Мотор-редуктор для работа 90° [Электронный ресурс] // URL: <https://www.yourduino.ru/product/motor-reduktor-dlya-robota-90>

28. ЧИП и ДИП - интернет-магазин приборов и электронных компонентов. AlphaBot, Платформа для создания мобильного робота на базе Raspberry Pi 3 Model B [Электронный ресурс] // URL: <https://www.chipdip.ru/product/alphabot>

29. ARDUINO-KIT | Электронные конструкторы и наборы, контроллеры, модули и датчики. Сервопривод TowerPro SG90 [Электронный ресурс] // URL: <https://arduino-kit.ru/product/servoprivod-towerpro-sg90>

30. СТО 4.2-07-2014 Система менеджмента качества. Общие требования к построению, изложению и оформлению документов учебной деятельности. – Введ. 30.12.2013. – Красноярск: ИПК СФУ, 2014. - 60с.

31. E.sfu-kras.ru – Электронные курсы СФУ [Электронный ресурс] // URL: <https://e.sfu-kras.ru/>

ПРИЛОЖЕНИЕ А

Сертификат за участие в IV Международной научно-практической конференции «SCIENCE AND EDUCATION: PROBLEMS AND INNOVATIONS».



ПРИЛОЖЕНИЕ Б

Диплом за 1 место в секции «Технические науки» IV Международной научно-практической конференции «SCIENCE AND EDUCATION: PROBLEMS AND INNOVATIONS».



ПРИЛОЖЕНИЕ В

Код программы передвижения робота, захвата изображения и передачи изображения в нейронную сеть для обработки.

```
import RPi.GPIO as GPIO
import time
from AlphaBot import AlphaBot

Ab = AlphaBot()

DR = 16
DL = 19

GPIO.setmode(GPIO.BCM)
GPIO.setwarnings(False)
GPIO.setup(DR, GPIO.IN, GPIO.PUD_UP)
GPIO.setup(DL, GPIO.IN, GPIO.PUD_UP)

i = 0 #Для счетчика фотографий
try:
    while True: #Бесконечный цикл считывания состояний датчиков и
        передвижения робота
            DR_status = GPIO.input(DR) #Считывание значения с правого
            датчика препятствий
            DL_status = GPIO.input(DL) #Считывание значения с левого датчика
            препятствий
            if((DL_status == 1) and (DR_status == 1)): #Если нет препятствий
                спереди
                    Ab.forward() #Движение вперед
                    time.sleep(2)
                    Ab.stop() #Остановка робота
                    time.sleep(2)
                    #Захват изображения и сохранения по указанному пути
                    camera.capture('/home/pi/Desktop/diplom/foto/' + str(i) +
                    '.jpg')
                    #Передача изображения в нейронную сеть для обработки
                    ns = 'python3 scripts/label_image.py --image ' + str(i) +
                    '.jpg'
                    os.system(ns)
                    i++
            elif((DL_status == 1) and (DR_status == 0)): #Если препятствие
                справа
                    Ab.left() #Движение влево
            elif((DL_status == 0) and (DR_status == 1)): #Если препятствие
                слева
```

```

        Ab.right() #Движение вправо
    else: #Иначе - препятствие спереди
        Ab.backward() #Движение назад
        time.sleep(0.2)
        Ab.left()
        time.sleep(0.2)
        Ab.stop() #Остановка работа

except KeyboardInterrupt:
    GPIO.cleanup();

```

Код программы обработки изображения нейронной сетью.

```

#Сортировка данных для вывода результатов в порядке убывания
top_k = results.argsort()[-5:][::-1]
labels = load_labels(label_file)

for i in top_k: #Обход вариантов дефектов
    if labels[i] == 'purely' and results[i] >= 0.60: #Если нет
дефекта
        remove = 'rm ' + file_name
        os.system(remove) #Удалить снимок
    elif results[i] >= 0.60: #Если есть дефект
        rename = 'mv ' + file_name + ' ' + labels[i] +
dt.datetime.now().strftime('%d-%m-%Y-%H-%M-%S') + '.jpg'
        os.system(rename) #Переименовать снимок
    else #Иначе - сохранить снимок (что-то новое на снимке)
        rename = 'mv ' + file_name + ' ' + 'new_defect' +
dt.datetime.now().strftime('%d-%m-%Y-%H-%M-%S') + '.jpg'
        os.system(rename) #Сохранить снимок

```

Федеральное государственное автономное
образовательное учреждение
высшего образования
«СИБИРСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»

Институт космических и информационных технологий

институт

Вычислительная техника

кафедра

УТВЕРЖДАЮ
Заведующий кафедрой

О. В. Непомнящий

подпись инициалы, фамилия

« » 2020 г.

МАГИСТЕРСКАЯ ДИССЕРТАЦИЯ

Интеллектуальная система управления змеевидным роботом

Тема

09.04.01 «Информатика и вычислительная техника»

код и наименование направления

09.04.01.06 «Микропроцессорные системы»

код и наименование магистерской программы

Научный
руководитель



профессор, зав кафедры, к.т.н

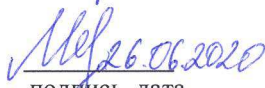
подпись, дата

должность, ученая степень

О.В. Непомнящий

инициалы, фамилия

Выпускник



подпись, дата

Е.А. Минин

инициалы, фамилия

Рецензент



подпись, дата

руководитель деп.инф.техн.

должность

А.С. Стаценко

инициалы, фамилия

Нормоконтролер



подпись, дата

О.В. Непомнящий

инициалы, фамилия

Красноярск 2020