

Министерство образования и науки Российской Федерации
федеральное государственное автономное образовательное учреждение
высшего образования
**«НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ
ТОМСКИЙ ПОЛИТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ»**

Школа Инженерная школа информационных технологий и робототехники
Направление подготовки 15.03.06 Мехатроника и робототехника
Отделение школы
(НОЦ) Отделение автоматизации и робототехники

БАКАЛАВРСКАЯ РАБОТА

Тема работы
Применение нейронных сетей в задаче вычисления дистанции по стереоизображению УДК 621.865.8-026.26.004.85

Студент

Группа	ФИО	Подпись	Дата
8Е41	Чугунов Роман Анварович		

Должность	ФИО	Ученая степень, звание	Подпись	Дата
Руководитель ВКР	Пякилля Борис Иванович	-		
Руководитель ООП	Мамонова Татьяна Егоровна	к.т.н., доцент		

КОНСУЛЬТАНТЫ:

По разделу «Финансовый менеджмент, ресурсоэффективность и ресурсосбережение»

Должность	ФИО	Ученая степень, звание	Подпись	Дата
Доцент ОСГН	Петухов Олег Николаевич	к.э.н., доцент		

По разделу «Социальная ответственность»

Должность	ФИО	Ученая степень, звание	Подпись	Дата
Ассистент ОКД	Авдеева Ирина Ивановна	-		

ДОПУСТИТЬ К ЗАЩИТЕ:

Должность	ФИО	Ученая степень, звание	Подпись	Дата
Руководитель ОАР	Леонов Сергей Владимирович	к.т.н.		

ПЛАНИРУЕМЫЕ РЕЗУЛЬТАТЫ ОБУЧЕНИЯ ПО ООП

Код результата	Результат обучения (выпускник должен быть готов)
<i>Профессиональные компетенции</i>	
P1	Применять глубокие естественно-научные, математические знания в области анализа, синтеза и проектирования для решения научных и инженерных задач производства и эксплуатации мехатронных и робототехнических устройств и систем, в том числе их систем управления
P2	Воспринимать, обрабатывать, анализировать и обобщать научно-техническую информацию, передовой отечественный и зарубежный опыт в области теории, проектирования, производства и эксплуатации мехатронных и робототехнических устройств и систем, принимать участие в командах по разработке и эксплуатации таких устройств и систем
P3	Применять полученные знания для решения инженерных задач при разработке, производстве и эксплуатации современных мехатронных и робототехнических устройств и систем (в том числе интеллектуальных) с использованием технологий мирового уровня, современных инструментальных и программных средств
P4	Определять, систематизировать и получать необходимую информацию в области проектирования, производства, исследований и эксплуатации мехатронных и робототехнических модулей, устройств и систем
P5	Планировать и проводить аналитические, имитационные и экспериментальные исследования для целей проектирования, производства и эксплуатации мехатронных и робототехнических средств и систем с использованием передового отечественного и зарубежного опыта, уметь критически оценивать полученные теоретические и экспериментальные данные и делать выводы
<i>Универсальные компетенции</i>	
P6	Интегрировать знания в области анализа, проектирования, производства и эксплуатации мехатронных и робототехнических устройств и систем со знаниями из смежных областей
P7	Понимать используемые современные методы, алгоритмы, модели и технические решения в мехатронике и робототехнике и знать области их применения, в том числе в автоматизированных производствах.
P8	Эффективно работать в профессиональной деятельности индивидуально и в качестве члена команды
P9	Владеть иностранным языком на уровне, позволяющем работать в интернациональной среде с пониманием культурных, языковых и социально-экономических различий
P10	Проявлять широкую эрудицию, в том числе знание и понимание современных общественных и политических проблем, демонстрировать понимание вопросов безопасности и охраны здоровья сотрудников, юридических аспектов, ответственности за инженерную деятельность, влияния инженерных решений на социальный контекст и окружающую среду.
P11	Следовать кодексу профессиональной этики и ответственности и международным нормам инженерной деятельности
P12	Понимать необходимость и уметь самостоятельно учиться и повышать квалификацию в течение всего периода профессиональной деятельности

Министерство образования и науки Российской Федерации
федеральное государственное автономное образовательное учреждение
высшего образования
**«НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ
ТОМСКИЙ ПОЛИТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ»**

Школа Инженерная школа информационных технологий и робототехники
Направление подготовки 15.03.06 Мехатроника и робототехника
Отделение школы (НОЦ) Отделение автоматизации и робототехники

УТВЕРЖДАЮ:
Руководитель ООП
_____ Мамонова Т.Е.
(Подпись) (Дата) (Ф.И.О.)

ЗАДАНИЕ
на выполнение выпускной квалификационной работы

В форме:

Бакалаврской работы

(бакалаврской работы, дипломного проекта/работы, магистерской диссертации)

Студенту:

Группа	ФИО
8Е41	Чугунов Роман Анварович

Тема работы:

Применение нейронных сетей в задаче вычисления дистанции по стереоизображению

Утверждена приказом директора (дата, номер)

Срок сдачи студентом выполненной работы:

ТЕХНИЧЕСКОЕ ЗАДАНИЕ:

Исходные данные к работе

(наименование объекта исследования или проектирования; производительность или нагрузка; режим работы (непрерывный, периодический, циклический и т. д.); вид сырья или материал изделия; требования к продукту, изделию или процессу; особые требования к особенностям функционирования (эксплуатации) объекта или изделия в плане безопасности эксплуатации, влияния на окружающую среду, энергозатратам; экономический анализ и т. д.).

Предметом исследования является алгоритм построения карт глубины по стереоизображению на основе нейронной сети. Готовая программа должна строить карту глубины по стереоизображению.

<p>Перечень подлежащих исследованию, проектированию и разработке вопросов</p> <p><i>(аналитический обзор по литературным источникам с целью выяснения достижений мировой науки техники в рассматриваемой области; постановка задачи исследования, проектирования, конструирования; содержание процедуры исследования, проектирования, конструирования; обсуждение результатов выполненной работы; наименование дополнительных разделов, подлежащих разработке; заключение по работе).</i></p>	<ol style="list-style-type: none"> 1. Провести обзор литературы 2. Спроектировать структуру программы с нейронной сетью. 3. Обучить нейронную сеть. 4. Провести тестирование нейронной сети.
<p>Перечень графического материала</p> <p><i>(с точным указанием обязательных чертежей)</i></p>	<ol style="list-style-type: none"> 1. Блок-схемы алгоритма программы.
<p>Консультанты по разделам выпускной квалификационной работы</p> <p><i>(с указанием разделов)</i></p>	
<p>Раздел</p>	<p>Консультант</p>
<p>Финансовый менеджмент, ресурсоэффективность и ресурсосбережение</p>	<p>Петухов Олег Николаевич, доцент ОСГН, к.э.н.</p>
<p>Социальная ответственность</p>	<p>Авдеева Ирина Павловна, ассистент ОКД</p>
<p>Названия разделов, которые должны быть написаны на русском и иностранном языках:</p>	
<p>Нет</p>	

<p>Дата выдачи задания на выполнение выпускной квалификационной работы по линейному графику</p>	<p>01.03.2018</p>
--	-------------------

Задание выдал руководитель:

Должность	ФИО	Ученая степень, звание	Подпись	Дата
<p>Старший преподаватель ОАР</p>	<p>Пякилля Б.И.</p>	<p>-</p>		<p>01.03.2018</p>

Задание принял к исполнению студент:

Группа	ФИО	Подпись	Дата
<p>8Е41</p>	<p>Чугунов Р.А.</p>		<p>01.03.2018</p>

Министерство образования и науки Российской Федерации
федеральное государственное автономное образовательное учреждение
высшего образования
**«НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ
ТОМСКИЙ ПОЛИТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ»**

Школа Инженерная школа информационных технологий и робототехники
Направление подготовки (специальность) 15.03.06 Мехатроника и робототехника
Уровень образования бакалавриат
Отделение школы
(НОЦ) Отделение автоматизации и робототехники
Период выполнения _____ (осенний / весенний семестр 2017/2018 учебного года)

Форма представления работы:

Бакалаврская работа

(бакалаврская работа, дипломный проект/работа, магистерская диссертация)

**КАЛЕНДАРНЫЙ РЕЙТИНГ-ПЛАН
выполнения выпускной квалификационной работы**

Срок сдачи студентом выполненной работы:

Дата контроля	Название раздела (модуля) / вид работы (исследования)	Максимальный балл раздела (модуля)
01.06.2018	Основная часть	60
10.05.2018	Финансовый менеджмент, ресурсоэффективность и ресурсосбережение	20
25.05.2018	Социальная ответственность	20

Составил преподаватель:

Должность	ФИО	Ученая степень, звание	Подпись	Дата
Старший преподаватель ОАР	Пякилля Б.И.	-		

СОГЛАСОВАНО:

Руководитель ООП	ФИО	Ученая степень, звание	Подпись	Дата
Доцент ОАР	Мамонова Т.Е.	к.т.н.		

Реферат

Работа содержит 92 страницы, 22 рисунка и 22 таблицы, а также два приложения: блок-схемы алгоритмов и исходных код программы.

Ключевые слова: стереоизображение, карта глубины, машинное обучение, Keras, Tensorflow.

Цель работы: разработать программную компоненту для построения карт глубины по стереоизображению.

В данной работе была разработана программная компонента на основе нейронной сети, которая позволяет строить карты глубины по стереоизображениям.

Был выполнен обзор существующих решений и теоретический обзор для решения задачи разработки программы. Была описана структура нейронной сети и процесс ее обучения.

Содержание

Введение	10
1 Обзор литературы	12
2 Объект и методы исследования	15
2.1 Машинное обучение	15
2.2 Машинное обучение	19
2.3 Машинное обучение	26
2.4 Машинное обучение	28
3 Разработка программной компоненты	29
3.1 Инструменты	29
3.2 Выборка	30
3.3 Разработка структуры нейронной сети.....	31
3.4 Процесс обучения нейронной сети	35
3.5 Блок-схема алгоритма обучения нейронной сети	36
4 Результаты проведенного исследования	36
5 Финансовый менеджмент, ресурсоэффективность и ресурсосбережение	39
5.1 Оценка коммерческого потенциала и перспективности проведения научных исследований с позиции ресурсоэффективности и ресурсосбережения	41
5.1.1 Потенциальные потребители результатов исследования	41
5.1.2 Анализ конкурентных технических решений	41
5.1.3 Технология QuaD	44
5.1.4 SWOT-анализ	46
5.1.5 Морфологический анализ	47
5.2 Планирование научно-исследовательских работ	48
5.2.1 Структура работ в рамках научного исследования	48
5.2.2 Определение трудоемкости выполнения работ	49
5.2.3 Разработка графика проведения научного исследования	52
5.2.4 Бюджет научно-технического исследования	54
5.2.5 Расчет материальных затрат	54
5.2.6 Основная заработная плата исполнителям темы	54
5.2.7 Дополнительная заработная плата	56
5.2.8 Отчисления во внебюджетные фонды	56
5.2.9 Накладные расходы	57
5.2.10 Контрагентные расходы	58

5.2.11 Формирование бюджета затрат научно-исследовательского проекта	58
5.2.12 Определение ресурсной, финансовой и экономической эффективности ресурсов	59
6 Социальная ответственность	63
Введение	63
6.1 Производственная безопасность	64
6.1.1 Освещение	65
6.1.2 Шум	66
6.1.3 Микроклимат	66
6.1.4 Статическое электричество	67
6.1.5 Психофизиологические факторы	68
6.1.6 Электрический ток	69
6.1.7 Заключение по разделу	71
6.2 Экологическая безопасность	72
6.3 3 Безопасность в чрезвычайных ситуациях	73
6.4 Правовые и организационные вопросы обеспечения безопасности	74
6.5 Заключение по разделу	75
Заключение	76
Список публикаций студента	78
Список использованных источников	79
Приложения	81
Приложение А. Блок-схема алгоритма	81
Приложения Б. Исходный код	83

Определения

Стереоскопическое зрение — вид зрения, при котором возможно восприятие формы, размеров и расстояния до предмета, например, благодаря бинокулярному зрению.

Карта глубины – изображение, содержащее информацию о дистанции до объектов на изображении.

Введение

При решении задачи управления роботом, особенно мобильным, зачастую используется информация о расстоянии до объектов, окружающих его. Для сбора такой информации могут использоваться различные датчики расстояния. Однако простые датчики расстояния позволяют собрать только разреженную информацию об окружающем пространстве. Получение информации о множестве точек вокруг робота требует использования дорогостоящих датчиков, например, лидаров. В то же время для построения карты расстояний до объектов может использоваться стереопара, по изображению с которой строится карта глубины.

Стереоскопическое зрение (от греч. *στερεός* — твёрдый, пространственный) — вид зрения, при котором возможно восприятие формы, размеров и расстояния до предмета, например, благодаря бинокулярному зрению.

Одна из важных задач стереозрения - процесс преобразования двух плоских изображений в трехмерную сцену с восстановлением информации о глубине каждой точки плоского изображения (расстоянии от стереокамеры до соответствующей точки реальной сцены).

Зная информацию о глубине (функция двух переменных) можно генерировать 3D модели ландшафта и других природных объектов для использования в различных приложениях, таких как виртуальная реальность, симуляция полета, робототехника. В частности, знание об удаленности точек изображения от реального прообраза позволяет делать захват опорных точек движущегося объекта для получения трехмерных координат, что важно в индустрии спецэффектов, кино и телевидении, однако также может использоваться в робототехнике. Кроме того, захват координат опорных точек позволяет строить уникальные бесконтактные человеко-машинные интерфейсы,

управление в которых осуществляется движениями пальцев рук или вообще жестов в пространстве на некотором удалении от стереокамеры.

Коммерческие продукты, позволяющие строить карты глубины, достаточно дороги, например, стоимость стереокамеры с функцией построения карты глубины ZED от компании StereoLabs составляет \$450.

На данный момент известны активные и пассивные методы восстановления информации о глубине реальной сцены. Активные методы используют ультразвуковые преобразователи или лазерное освещение рабочего пространства, дающие на выходе быструю и точную информацию о глубине. Однако у этих методов есть ограничения по отношению к диапазону измерений и стоимости аппаратных компонентов.

Пассивные методы, основанные на компьютерном зрении, обычно реализуются более простыми и недорогими сенсорами, определяющими расстояние. Такие методы способны генерировать информацию о глубине по полученной паре изображений и параметрам двух камер. Одна из фундаментальных проблем стереозрения заключается в установлении точного соответствия между левым и правым изображением стереопары. Под соответствием понимается расстояние (диспаратет) между пикселями одного и того же объекта на левом и правом изображении. Существующие локальные методы по вычислению карты диспаратета основаны на принципе «скользящего окна». Они характеризуются определенной сбалансированностью скорости-качества, хорошо масштабируются и реализуются на таких параллельных архитектурах, как видеокарты.

Карты глубины широко применяются в задачах робототехники, в частности, в задаче SLAM (одновременное построение карты и локализация робота). Например, в [1] карты глубины используются для навигации антропоморфного робота.

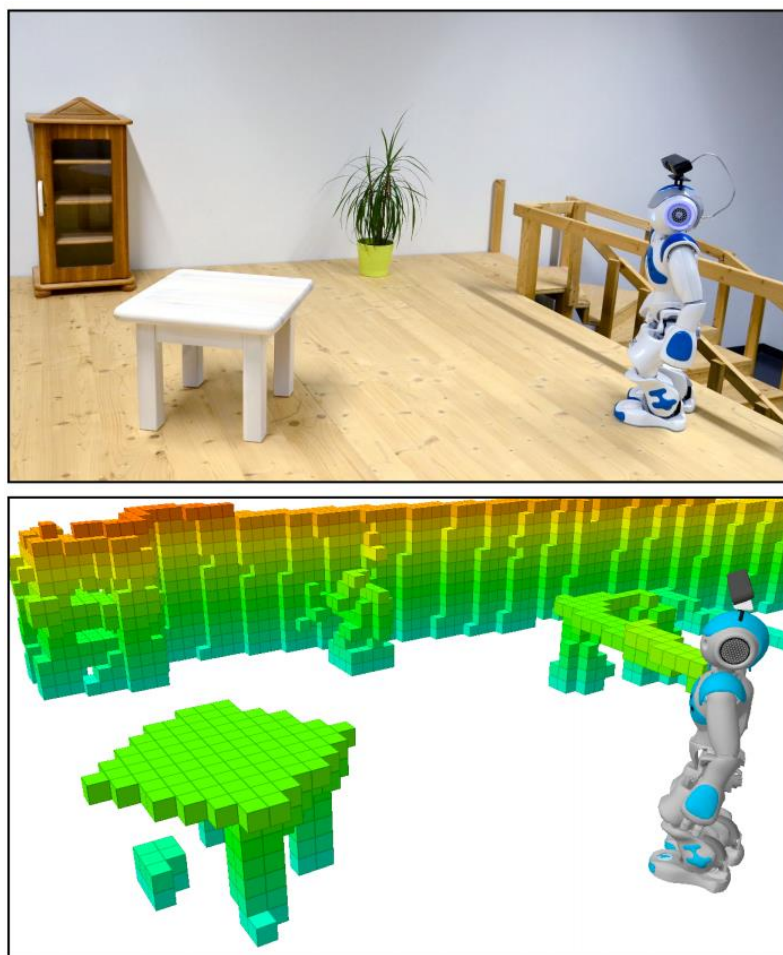


Рисунок 1. Навигация антропоморфного робота с помощью карты глубины

Важно рассчитывать позиции объектов в видимой роботом области, чтобы детектировать препятствия, планировать движение и локализовать робота. Данную информацию можно получить с помощью построения карты глубины.

Целью данной работы является создание алгоритма построения карт глубины (изображений, хранящих информацию о расстоянии между камерой и объектами) по стереоизображению.

1. Обзор литературы

Основной проблемой в использовании систем стереозрения является определение пар пикселей изображений с разных камер, которые относятся к одной и той же точке в пространстве.

В целом, алгоритмы, решающие данную проблему, могут быть разбиты на два класса [2]:

- Локальные методы
- Глобальные методы

Глобальные методы сводятся к минимизации так называемого энергетической функции между пикселями. Большинство данных методов вычислительно дороги.

Локальные методы вычислительно проще, поэтому больше подходят для построения карты глубины в реальном времени.

В основном, существуют следующие методы построения карт глубины:

Таблица 1. Методы построения карты глубины.

Метод	Описание
Вергенция	Оптические оси должны пересекаться на объекте, тогда можно совместить пиксели, относящиеся к одной точки поверхности
Диспаратность	Глубина рассчитывается по несоответствию левого и правого изображения. Считается, что у близких объектов несоответствий больше между левым и правым изображением.
Использование информации о размере объектов или их расположении	Зная размер объектов или расстоянии между ними на изображении можно определить расстояние до них.

На практике чаще всего используются метод диспаратности, который заключается в построения изображения, отражающего различия между левым и правым изображением, а также метод стереосовмещения.

Другие методы используются более редко в связи с ограничениями в применении, накладываемыми спецификой данных способов построения карт глубины.

На рисунке 2 можно увидеть пример диспаратного изображения, построенного в Matlab.

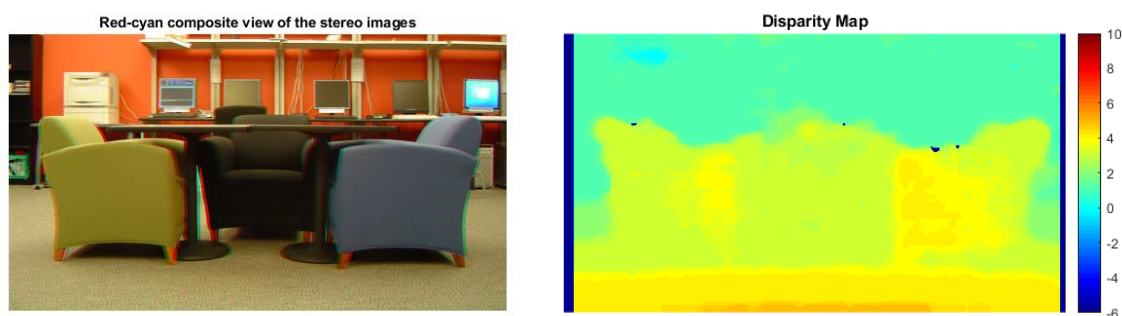


Рисунок 2. Диспаратное изображение, полученное в Matlab

На рисунке 3 можно увидеть пример диспаратного изображения, построенного с помощью библиотеки компьютерного зрения OpenCV.

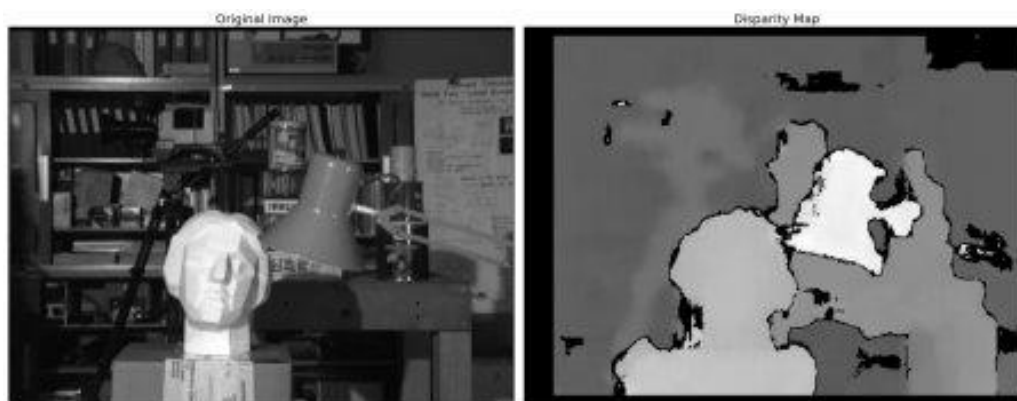


Рисунок 3. Диспаратное изображение, построенное с помощью библиотеки OpenCV

В последние годы все чаще для решения задачи построения карты глубины по стереоизображению используют методы машинного обучения, в частности методы глубокого обучения.

Глубокое обучение основано на использовании нейронных сетей. Данный метод хорошо зарекомендовал себя в компьютерном зрении. Использование нейронных сетей в задаче построения карт глубины может позволить добиться нового уровня качества.

2. Объект и методы исследования

2.1 Машинное обучение

Машинное обучение является частью науки о данных. Слово обучение в данном термине означает, что алгоритмы машинного обучения зависят от некоторого набора данных, которые используются для тренировки модели, валидации и теста.

Машинное обучение включает в себя множество методов, подходящих для решения различных задач. В машинном обучении выделяют три основные области:

- Обучение с учителем (supervised learning) — наиболее распространённый случай. Каждый обучающий пример (прецедент) представляет собой пару «объект, ответ».
- Обучение без учителя (unsupervised learning) — в этом случае ответы не задаются и требуется искать зависимости между объектами.
- Обучение с подкреплением (reinforcement learning) — роль объектов играют пары «ситуация, принятое решение», ответами являются значения функционала качества, характеризующего правильность принятых решений (реакцию среды).

На рисунке 4 представлены категории методов, которые включает в себя машинное обучение, а также области, в которых данные методы могут применяться.

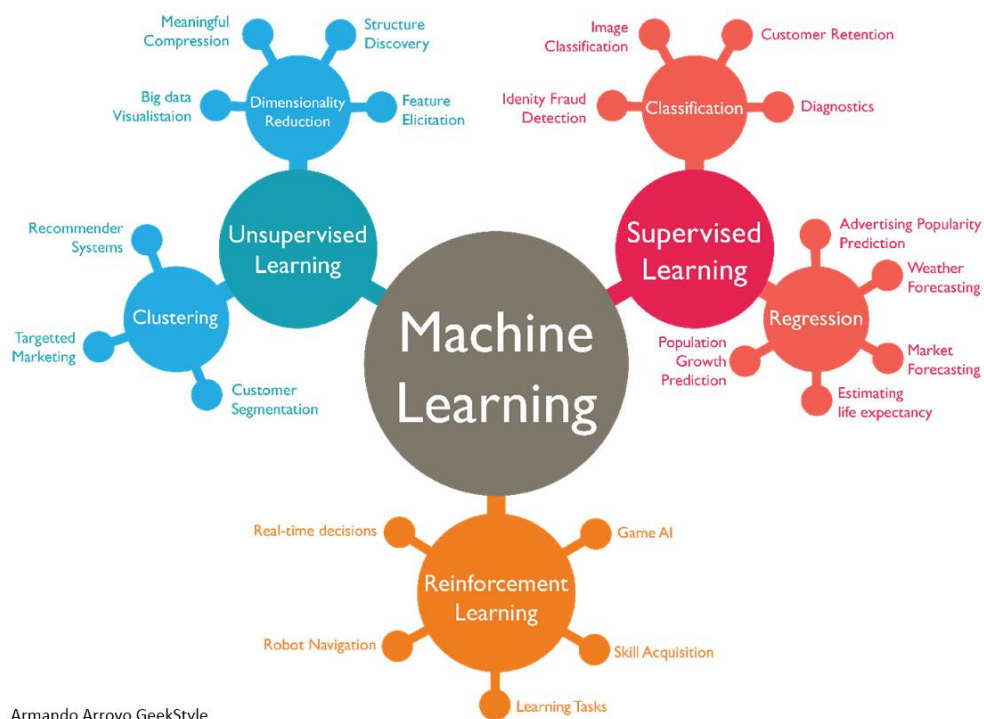


Рисунок 4. Области машинного обучения

В данном случае используемая модель машинного обучения относится к классу алгоритмов обучения с учителем. Основные задачи, которые могут решать методы данного класса:

- Задача классификации (classification) отличается тем, что множество допустимых ответов конечно. Их называют метками классов (class label). Класс — это множество всех объектов с данным значением метки.
- Задача регрессии (regression) отличается тем, что допустимым ответом является действительное число или числовой вектор.
- Задача ранжирования (learning to rank) отличается тем, что ответы надо получить сразу на множестве объектов, после чего отсортировать их по значениям ответов. Может сводиться к задачам классификации или регрессии. Часто применяется в информационном поиске и анализе текстов.

- Задача прогнозирования (forecasting) отличается тем, что объектами являются отрезки временных рядов, обрывающиеся в тот момент, когда требуется сделать прогноз на будущее. Для решения задач прогнозирования часто удаётся приспособить методы регрессии или классификации, причём во втором случае речь идёт скорее о задачах принятия решений.

Так как решаемая задача относится к классу задач обучения с учителем, следует упомянуть, какие алгоритмы обычно используются при решении таких задач.

Суть задачи классификации заключается в том, что программа должна относить новые данные к некоторому классу. Классов может быть два (определение пола человека, спам/не спам сообщение и т.д.) и больше. Каждый объект может принадлежать одному или, в некоторых случаях, нескольким классам одновременно.

Суть задачи регрессии заключается в том, что программа должна рассчитывать значение некоторой непрерывной функции от признаков объектов. Задачи регрессии и классификации связаны между собой, алгоритм регрессии можно преобразовать к алгоритму классификации путем дискретизации выходных значений. Поэтому различные алгоритмы зачастую могут быть использованы и для регрессионного анализа, и для задачи классификации с минимальными доработками.

Некоторые примеры задач классификации: распознавание речи, распознавание рукописных символов, биометрическая идентификация, классификация документов и т.д.

Примеры задач регрессии: оценка рисков выдачи кредита заёмщику, предсказание стоимости чего-то либо, управление скоростями приводов робота в зависимости от измеренных датчиками значений.

К основным алгоритмам можно отнести:

- Линейные классификаторы: логистическая регрессия, наивный байесовский классификатор
- Метод опорных векторов
- Решающие деревья
- Метод градиентного бустинга над решающими деревьями
- Случайный лес
- Нейронные сети
- Метод ближайшего соседа (а также метод ближайших соседей)

Ниже приведено краткое описание данных методов.

Наивный байесовский классификатор – метод классификации, основанный на теореме Байеса с предположением о независимости среди признаков (предикторов) объекта выборки. Естественно, что на практике это лишь частный случай, зачастую одни признаки объектов связаны с другими. Однако, даже если данные признаки зависят друг от друга, предполагается, что все они независимо друг от друга влияют на вероятность принадлежности к какому-либо классу. Такую модель легко построить, кроме того, она показывает хорошие результаты на больших наборах данных.

Логистическая регрессия – статистический метод для анализа данных, в которых несколько независимых переменных определяют зависимую. Как правило, метод используется в бинарной классификации, однако также может использоваться в мультиклассификации.

Решающие деревья – данный метод строит модель классификации или регрессии в виде дерева. Строящееся дерево разбивает выборку во все меньшие и меньшие подвыборки. Решающие деревья могут работать с категориальными, так и численными данными.

Случайный лес – ансамбль (определенным образом построенная совокупность моделей) решающих деревьев. Особенность данного метода

заключается в том, что каждое дерево в ансамбле строится только на некоторой подвыборке исходной обучающей выборки.

Нейронная сеть – нейронная сеть состоит из нейронов в слоях, которые переводят входной вектор в некоторый выход. В общем случае каждый нейрон принимает на вход некоторое значение, применяет нелинейную (чаще всего) функцию активации и передает на следующий слой.

Метод ближайших соседей – метод, не требующий обучения. Новые данные относятся к какому-либо классу следующим способом: по некоторой функции расстояния определяется дистанция до объектов «обучающей» выборки, затем объект относится к классу, которому принадлежит большинство из k (нечетное число) ближайших объектов. В качестве функции расстояния может применяться, например, евклидово расстояние.

2.2 Глубокое обучение

Глубокое обучение (глубинное обучение; англ. Deep learning) — совокупность методов машинного обучения (с учителем, с частичным привлечением учителя, без учителя, с подкреплением), основанных на обучении представлениям (англ. feature/representation learning), а не на специализированных алгоритмах, разработанных для конкретных задач. Многие методы глубокого обучения были известны ещё в 1980-е (и даже ранее), но результаты были невпечатляющими, пока продвижения в теории искусственных нейронных сетей (предобучение нейросетей с помощью специального случая ненаправленной графической модели, так называемой ограниченной машины Больцмана) и вычислительные мощности середины 2000-х (прежде всего, графических процессоров Nvidia, а в настоящее время и тензорных процессоров Google) не позволили создавать сложные технологические архитектуры нейронных сетей, обладающие достаточной производительностью и позволяющие решать широкий спектр задач, не поддававшихся эффективному решению ранее, например, в компьютерном зрении, машинном переводе,

распознавании речи, причём качество решения во многих случаях теперь сопоставимо, а в некоторых случаях превосходит эффективность «белковых» экспертов. [3]

На рисунке 5 можно увидеть отличия между традиционным машинным обучением и глубоким обучением. Основное отличие в том, что этап извлечения признаков не столь важен в глубоком обучении, как в других методах.

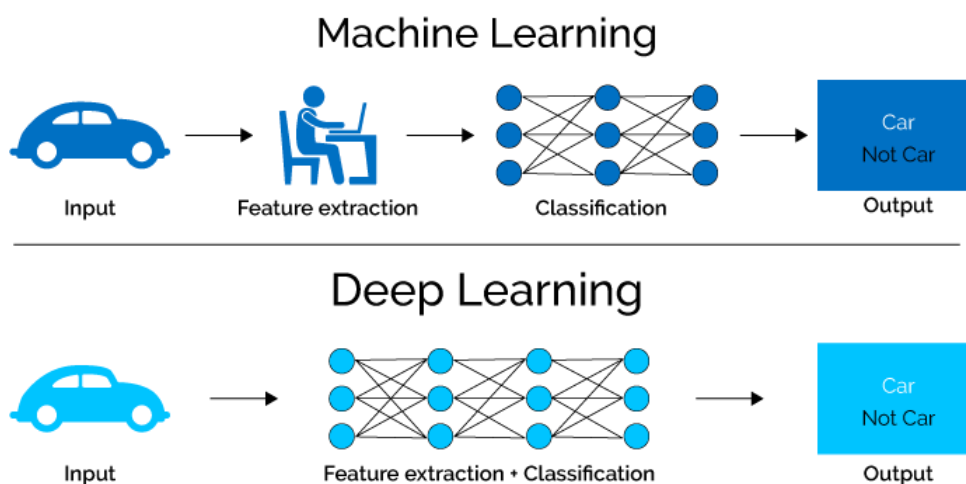


Рисунок 5. Отличия между машинным обучением и глубоким обучением

Нейронные сети предсказывают выходные значения, получая данные на вход.

Нейронную сеть определяют такие компоненты, как:

- Искусственные нейроны
- Топология сети – способ соединения нейронов
- Веса
- Алгоритм обучения

Структура искусственного нейрона зависит от выбранного типа (архитектуры) нейронной сети.

Целью алгоритма обучения является решение задачи оптимизации весов нейронной сети таким образом, чтобы ошибка предсказаний на обучающей

выборке была минимальной. Ошибка определяется согласно установленной метрике качества.

Таким образом, перед тем, как нейронная сеть может точно предсказывать результат на некоторых новых данных, её нужно обучить на обучающей выборке. Обучающие данные как правило состоят из пар вход-выход (в случае обучения с учителем). Выходы также называют метками или целевыми переменными

Фаза обучения начинается со случайной инициализации всех весов (каждый вес соответствует какому-либо искусственному нейрону). Затем входные данные подаются на первый слой нейронной сети, в соответствии с типом нейрона вычисляются их выходы, к ним применяется функция активации и результаты передаются дальше к другим слоям. Этот процесс называют прямым распространением. Так как веса заданы случайно, результат на выходе нейронной сети неправилен. Целью является прибыть в ту точку в пространстве весов, в котором нейронная сеть может предсказывать выходные значения с достаточной точностью. Алгоритм такой настройки весов нейронной сети называется обратным распространением ошибки.

Однако существуют и другие алгоритмы, которые можно применить для оптимизации весов нейронной сети – наиболее известными среди них являются метод роя частиц и генетические алгоритмы.

После прямого распространения на выходе получены некоторые результаты, которые в идеальном случае должны соответствовать целевым переменным. Однако, в начале обучения это не так, ведь веса были заданы случайно. По некоторой функции ошибок рассчитывается ошибка, которая «распространяется» по сети в обратном направлении. В качестве метода минимизации функции ошибок используется оптимизационный метод градиентного спуска.

Пример графика функции ошибок в пространстве весов можно увидеть на рисунке 6.

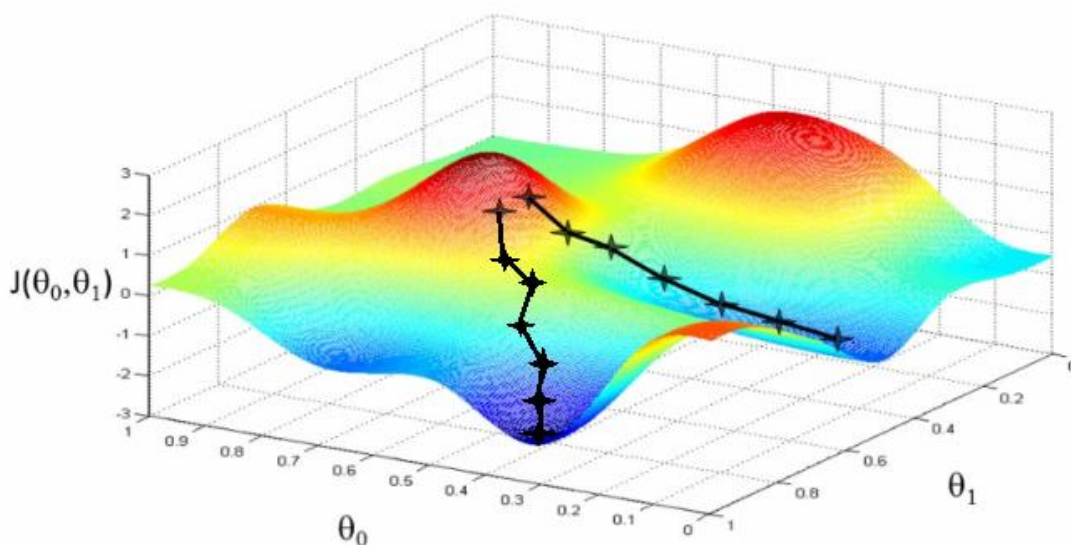


Рисунок 6. Пример графика функции ошибок

Идея градиентного спуска состоит в совершении небольших шагов по градиенту пока не достигнут минимум (локальный или глобальный). Так как функция ошибок чаще всего имеет сложную поверхность, в результате обучения достигается только локальный минимум. Таким образом, немного изменяя веса на каждой итерации обучения, мы движемся по градиенту. Ширина шага зависит от параметра, называемого коэффициентом скорости обучения (learning rate). Данный параметр не должен быть слишком большим, так как в таком случае обучение не сойдется к минимуму, но и не может быть слишком маленьким, так как в таком случае обучение займет слишком много времени (слишком много эпох обучения).

На практике чаще используются модификации классического градиентного спуска, например, Adam (Adaptive moment estimation). Данный оптимизационный алгоритм сочетает в себе идею накопления движением (импульс) и идею уменьшенного обновления весов для типичных признаков. На рисунке 7 приведен пример работы различных оптимизационных алгоритмов, запущенных в одной и той же начальной точке.

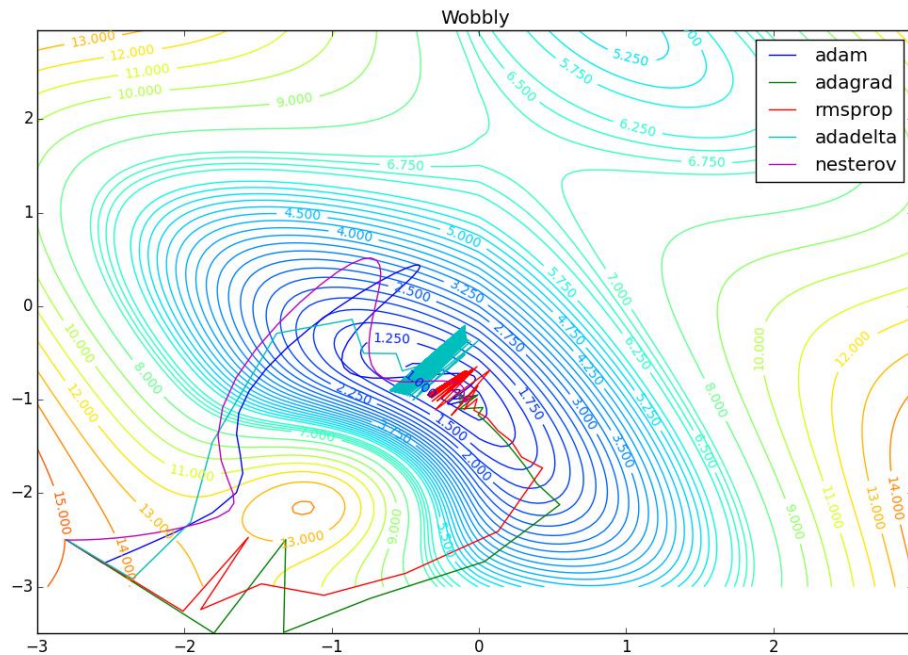


Рисунок 7. Результат работы различных оптимизационных алгоритмов

Существует множество различных архитектур нейронных сетей. Вот список основных из них:

- Сеть прямого распространения – каждый нейрон соединен со всеми нейронами предыдущего и следующего слоев, активации выполняются последовательно от первого к последнему слою, без обратных связей, один скрытый слой.
- Глубокая сеть прямого распространения – сеть прямого распространения с несколькими скрытыми слоями.
- Рекуррентная нейронная сеть – нейронная сеть с рекуррентными нейронами. В первой нейронной сети такого типа нейроны получали выход с задержкой в несколько итераций.
- Сеть долгой краткосрочной памяти – нейронная сеть, состоящая из ячеек памяти, предназначенных обрабатывать данные, которые распределены во времени. Такие сети зачастую используются при работе с текстовой и звуковой информацией, с различными временными рядами, а также видео.

- Сеть с управляемыми рекуррентными нейронами – такая сеть очень похожа на сеть долгой краткосрочной памяти, слабо отличима по качеству работы, однако отличается сниженными требованиям к ресурсам.
- Автоэнкодер – сеть, чей выход совпадает со входом, что позволяет обучать её без учителя. Обычно используется для получения сжатого представления данных в пространстве признаков, однако также может использоваться для заполнения пропусков в данных.
- Глубокая сверточная сеть – сеть, имеющая несколько сверточных слоев. Обычно такая сеть имеет несколько слоев прямого распространения, однако иногда, например, в задачах сегментации изображений, сети могут быть полностью сверточными.
- Генеративно-сопоставительная сеть – сеть, представляющая собой совокупность двух нейронных сетей (генератора и дискриминатора). Дискриминатор получает обучающие или созданные генератором данные. Степень угадывания дискриминатором источника данных в дальнейшем участвует в формировании ошибки.
- Капсульная (в некоторых источниках капсульная) нейронная сеть – особенностью данного вида нейронных сетей является наличие так называемых капсул – промежуточных элементов между нейронами и слоями. Данная архитектура призвана исправить определенные недостатки сверточных сетей. Архитектура была задумана Джеффри Хинтоном в 1979 году, сформулирована в 2011 и опубликована в двух статьях в октябре 2017 года [4].

В данном случае будет использоваться сверточная сеть. В обычном перцептроне, который представляет собой полносвязную нейронную сеть, каждый нейрон связан со всеми нейронами предыдущего слоя, причём каждая связь имеет свой персональный весовой коэффициент. В свёрточной нейронной сети в операции свёртки используется лишь ограниченная матрица весов небольшого размера, которую «двигают» по всему обрабатываемому слою

(в самом начале — непосредственно по входному изображению), формируя после каждого сдвига сигнал активации для нейрона следующего слоя с аналогичной позицией. То есть для различных нейронов выходного слоя используются одна и та же матрица весов, которую также называют ядром свёртки. Её интерпретируют как графическое кодирование какого-либо признака, например, наличие наклонной линии под определённым углом. Тогда следующий слой, получившийся в результате операции свёртки такой матрицей весов, показывает наличие данного признака в обрабатываемом слое и её координаты, формируя так называемую карту признаков. Естественно, в сверточной нейронной сети набор весов не один, а целая гамма, кодирующая элементы изображения (например, линии и дуги под разными углами). При этом такие ядра свёртки не закладываются исследователем заранее, а формируются самостоятельно путём обучения сети классическим методом обратного распространения ошибки. Проход каждым набором весов формирует свой собственный экземпляр карты признаков, делая нейронную сеть многоканальной (много независимых карт признаков на одном слое). Также следует отметить, что при переборе слоя матрицей весов её передвигают обычно не на полный шаг (размер этой матрицы), а на небольшое расстояние. Так, например, при размерности матрицы весов 5×5 её сдвигают на один или два нейрона (пикселя) вместо пяти, чтобы не «перешагнуть» искомый признак.

Операция субдискретизации (англ. *subsampling*, англ. *pooling*, также переводимая как «операция подвыборки» или операция объединения), выполняет уменьшение размерности сформированных карт признаков. В данной архитектуре сети считается, что информация о факте наличия искомого признака важнее точного знания его координат, поэтому из нескольких соседних нейронов карты признаков выбирается максимальный и принимается за один нейрон уплотнённой карты признаков меньшей размерности. За счёт данной операции, помимо ускорения дальнейших вычислений, сеть становится более инвариантной к масштабу входного изображения.

Типовая сверточная сеть состоит из большого количества слоёв. После начального слоя (входного изображения) сигнал проходит серию свёрточных слоёв, в которых чередуется собственно свёртка и субдискретизация (пулинг). Чередование слоёв позволяет составлять «карты признаков» из карт признаков, на каждом следующем слое карта уменьшается в размере, но увеличивается количество каналов. На практике это означает способность распознавания сложных иерархий признаков. Обычно после прохождения нескольких слоёв карта признаков вырождается в вектор или даже скаляр, но таких карт признаков становятся сотни. На выходе свёрточных слоёв сети дополнительно устанавливают несколько слоёв полносвязной нейронной сети (перцептрон), на вход которому подаются окончательные карты признаков. В некоторых задачах слои полносвязной нейронной сети не добавляются (в частности, в задаче сегментации).

На рисунке 8 представлена типовая архитектура сверточной нейронной сети.

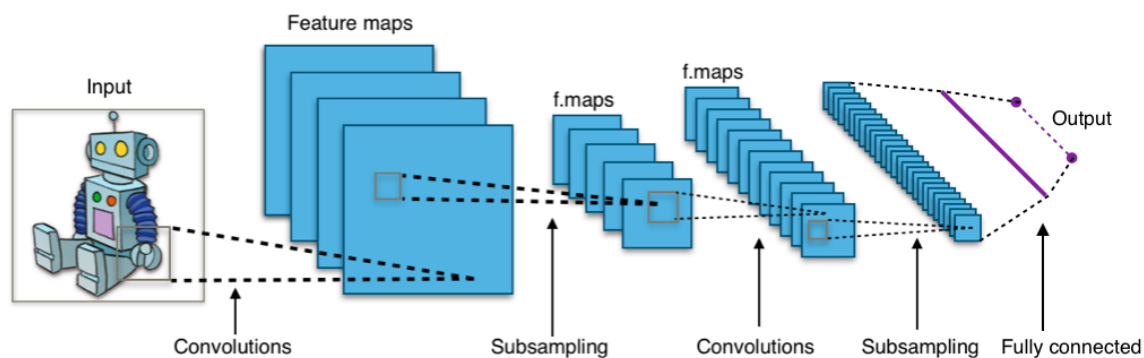


Рисунок 8. Типовая архитектура сверточной нейронной сети

2.3 Этапы работы с алгоритмом машинного обучения

Работу с алгоритмом машинного обучения с учителем можно разбить на два стадии: обучение и использование обученной модели. Чаще всего данные стадии включают в себя следующие шаги:

Обучение:

1. Сбор обучающих данных (обучающей выборки).

2. Предобработка и векторизация обучающих данных.
3. Разбиение выборки на 2 или более частей (обучающая, валидационная и тестовая)
4. Построение моделей машинного обучения.
5. Обучение моделей на обучающей выборке.
6. Валидация (проверка) на части выборки, не использовавшейся в обучении.
7. Шаги 4-6 повторяются, пока не отобраны лучшие модели.
8. Проверка отобранных моделей на тестовой выборке.

Использование обученной модели:

1. Получение новых данных
2. Предобработка и векторизация данных.
3. Предсказание на новых данных с помощью алгоритма машинного обучения.
4. Если качество предсказаний больше не отвечает установленным требованиям к уровню качества, алгоритм можно переобучить.

Целью сбора обучающих данных является формирование выборки, на которой можно будет обучить алгоритм. В различных задачах этот процесс происходит по-разному.

Предобработка данных очень важна в машинном обучении. Реальные наборы данных очень часто имеют следующие недостатки:

- Данные хранятся в неподходящем формате или в базе данных
- Данных может быть значительно больше, чем требуется
- В данных могут быть пропуски
- Отсутствие структуры в данных
- Признаки объектов в исходных данных малополезны, однако преобразования (извлечение признаков) может создать более полезные признаки

- Признаки зашумлены, имеются выбросы и сдвиги
- Признаки отличаются по порядку, что усложняет задачу обучения большинству алгоритмов

Поэтому предобработка данных должна включать в себя заполнение (или отбрасывание) пропусков, фильтрацию шума, структуризацию (в случае, если данные представляют собой неструктурированную информацию) и векторизацию (представление данных в матричном виде).

Векторизация изображений не требуется, так как изображения изначально представляют собой матрицы, которые требуется только считать правильным образом, однако может потребоваться совершить преобразования полученных матриц к нужной размерности или перевести в другое пространство (например, обработать изображения путем преобразования Фурье с целью получения новых признаков).

Зачастую применяются следующие способы предобработки данных:

- Стандартизация – каждый признак преобразуется таким образом, что среднее значение данного признака на всей выборке становится равным 0, а среднеквадратическое отклонение равняется 1.
- Нормализация признака в диапазоне (0, 1). Выполняется по формуле 1.

$$X_{norm} = \frac{X - X_{min}}{X_{max} - X_{min}}, \quad (1)$$

2.2 Метрики качества

Существуют различные методы оценки качества работы алгоритмов машинного обучения. Ниже представлены некоторые из методов на примере бинарной классификации:

- Точность (precision) - доля объектов, правильно отнесенных к этому классу из всех объектов, отнесенных к этому классу.

- Полнота (recall) - доля правильно классифицированных объектов данного класса из всех объектов данного класса.
- F-мера (F-score) - гармоническое среднее точности и полноты.
- ROC-AUC - площадь под кривой ошибок, строящейся в осях ложных положительных ответов и правильных положительных ответов.
- Среднеквадратическая ошибка (MSE)
- Корень из среднеквадратической ошибки (RMSE)

На рисунке 9 представлено графическое описание точности и полноты.

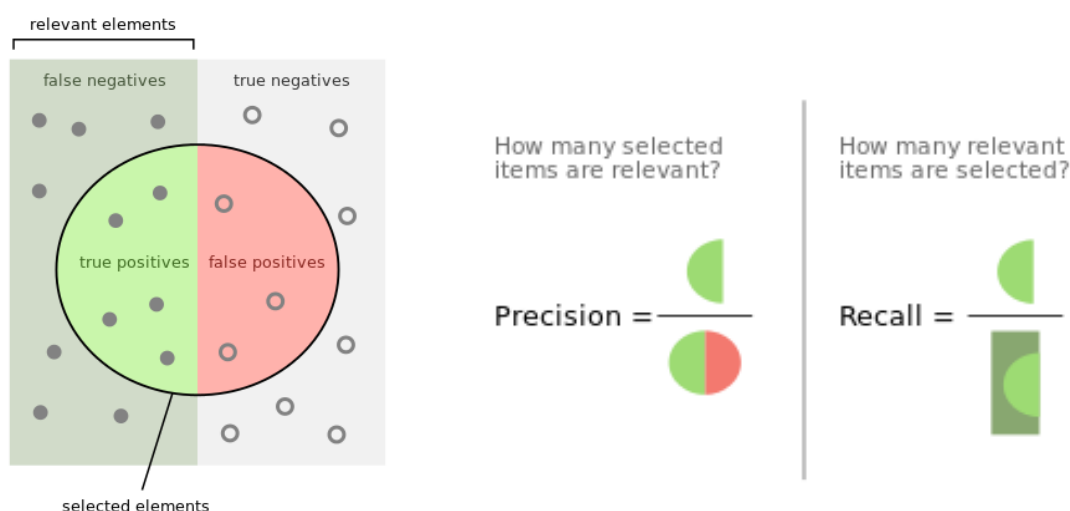


Рисунок 9. Графическое описание точности и полноты

3. Разработка программной компоненты

3.1 Инструменты

Самыми удобными языками программирования для разработки и прототипирования алгоритмов машинного обучения считаются Python и R. Это объясняется не только простотой синтаксиса, но и наличием большого числа открытых библиотек и фреймворков, позволяющих вести разработку на более высоком уровне.

В данном случае разработка ведется на Python, а используемыми библиотеками будут Tensorflow (библиотека машинного обучения) и Keras (библиотека для разработки и обучения нейронных сетей). Причина выбора

такого набора инструментов в том, что с ними уже имеется большой опыт работы.

3.2 Выборка

Для генерации выборки изображений для обучения используется программа для создания трехмерной компьютерной графики Blender. Разумеется, полученная выборка изображений не похожа на те изображения, которые как правило видит робот, однако достаточно большую и удобную базу данных стереоизображений реальных объектов с измеренными картами глубины получить не удалось. Поэтому выборка генерируется специально.

Blender — свободное и открытое программное обеспечение для создания трёхмерной компьютерной графики, включающий в себя средства моделирования, анимации, рендеринга, постобработки и монтажа видео со звуком, компоновки с помощью «узлов» (Node Compositing), а также для создания интерактивных игр. В настоящее время пользуется наибольшей популярностью среди бесплатных 3D-редакторов в связи с его быстрым и стабильным развитием, которому способствует профессиональная команда разработчиков.

Для генерации выборки был написана программа на языке Python, автоматически создающая стереоизображения и соответствующие им карты глубины на основе сгенерированной 3D сцены в Blender.

На рисунке 10 приведен интерфейс программы Blender.

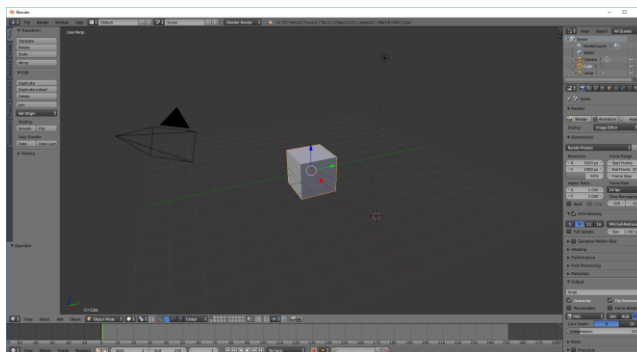


Рисунок 10. Интерфейс программы Blender

Обучающая выборка состоит из стереоизображений (рисунок 11) и карт глубины (рисунок 12).

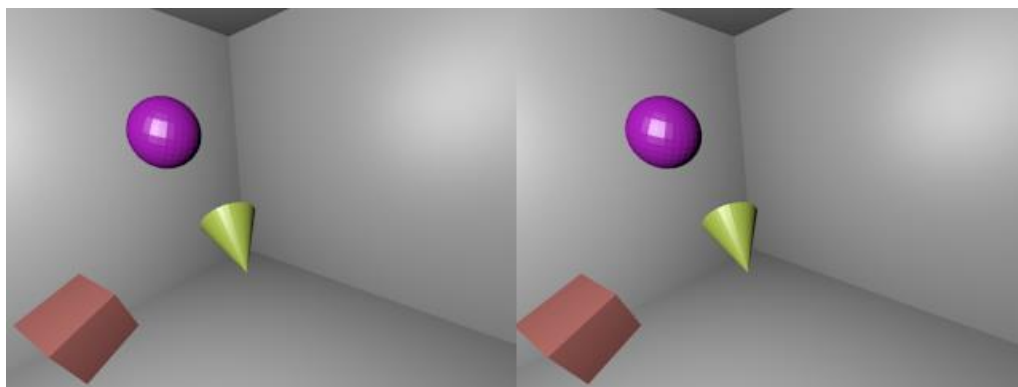


Рисунок 11. Пример стереоизображения

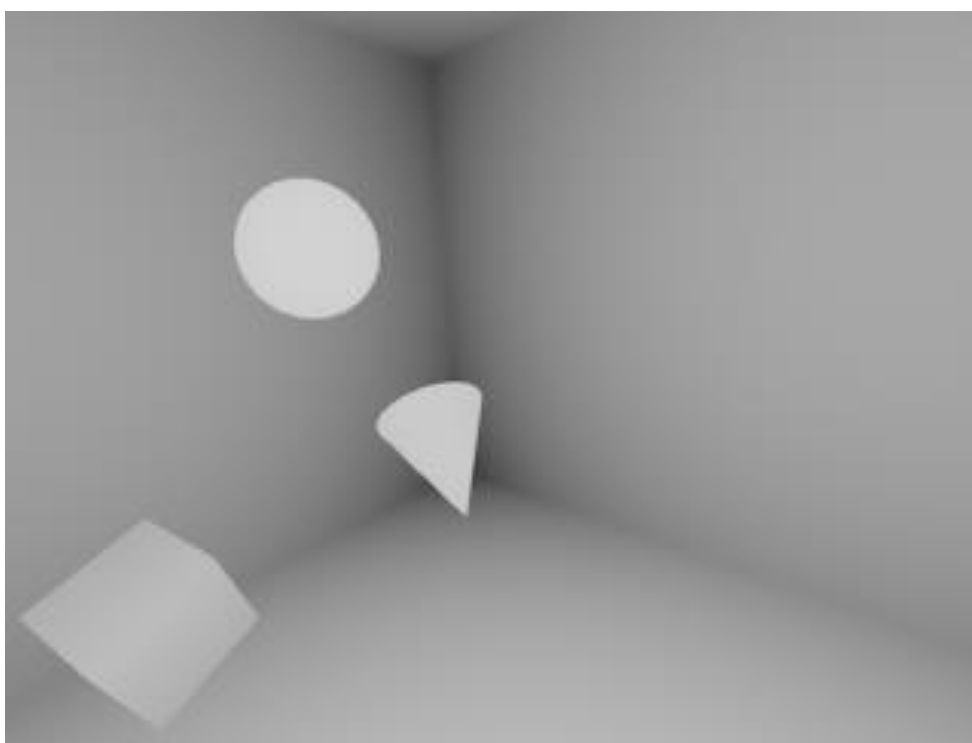


Рисунок 12. Пример карты глубины

3.3 Разработка структуры нейронной сети

Исходя из поставленной задачи построения карт глубины по стереоизображению, наиболее подходящей архитектурой сети являются сверточные сети, так как они, с одной стороны, хорошо подходят для обработки изображений, и, с другой стороны, не так требовательны к ресурсам, как капсульные сети.

Так как на выходе нейронной сети должно быть изображение, сеть будет полностью сверточной, т.е. без слоев других архитектур нейронных сетей. Это объясняется тем, что полностью сверточные сети эффективно работают в ситуациях, когда выходом является изображение. Такой подход часто применяется в задачах сегментации, где входное изображение преобразуется в маску, где каждый пиксель на выходе может иметь одно из дискретных значений – индекс класса, к которому относится данный объект изображения.

На вход подаются два изображения. В такой ситуации можно объединить изображения на входе и обрабатывать их как единое целое. Объединение изображений можно выполнить путем простой конкатенации матриц изображений.

На рисунке 13 представлена схема структуры нейронной сети. На данной схеме введены следующие обозначения:

- Св1 – сверточный слой с 16 фильтрами
- Суб1 – слой субдискретизации с окном 2x1
- Суб2 – слой субдискретизации с окном 2x2
- Св2 – сверточный слой с 32 фильтрами
- Св3 – сверточный слой с 64 фильтрами
- Св4 – сверточный слой с 128 фильтрами
- Св5 – сверточный слой с 256 фильтрами
- ТСв1 – слой транспонированной свертки с 128 фильтрами
- Кон – слой конкатенации
- ТСв2 – слой транспонированной свертки с 64 фильтрами
- ТСв3 – слой транспонированной свертки с 32 фильтрами
- ТСв4 – слой транспонированной свертки с 16 фильтрами

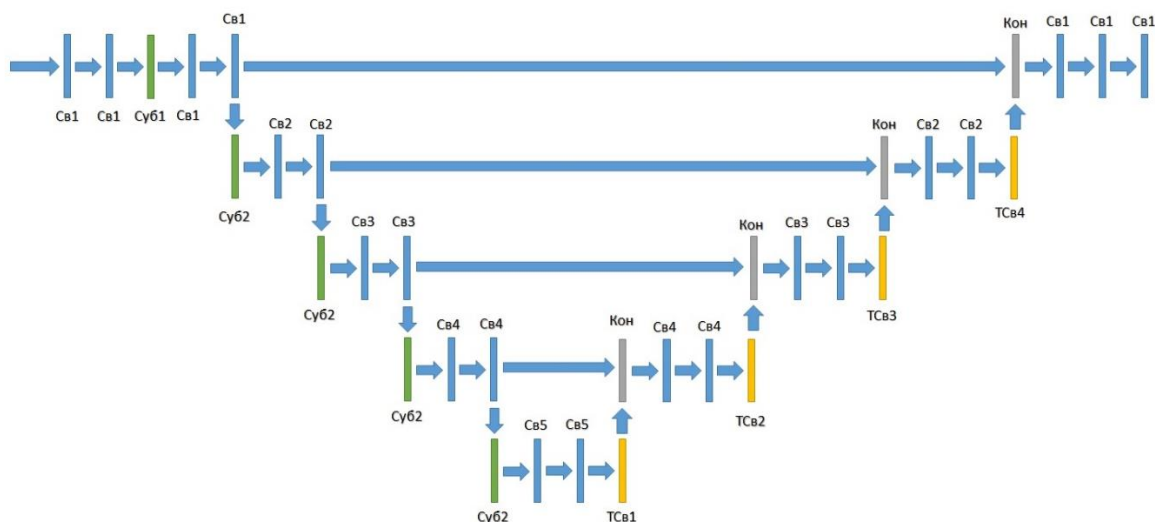


Рисунок 13. Структура нейронной сети

В структуре не указано, однако после каждого слоя нейронной сети добавлен слой *batch normalization*, предназначенный для ускорения обучения.

В нейронной сети такой структуры изображение размера $240 \times 640 \times 3$ (высота/ширина/3 канала RGB) меняет число своих пикселей следующим образом:

- До слоя Суб1: $240 \times 640 \times 16$ (16 фильтров после свертки)
- После слоя Суб1: $240 \times 320 \times 16$
- После субдискретизации на уровне 2: $120 \times 160 \times 16$
- После субдискретизации на уровне 3: $60 \times 80 \times 32$
- После субдискретизации на уровне 4: $30 \times 40 \times 64$
- После субдискретизации на уровне 5: $15 \times 20 \times 128$
- После ТСв1: $30 \times 40 \times 128$
- После Кон: $30 \times 40 \times 256$
- После ТСв2: $60 \times 80 \times 64$
- После Кон: $60 \times 80 \times 128$
- После ТСв3: $120 \times 160 \times 32$
- После Кон: $120 \times 160 \times 64$
- После ТСв4: $240 \times 320 \times 16$

- После Кон: 240x320x32

Слой транспонированной свертки представляет собой тоже свертку, однако она выполняется таким образом, что выходное изображение имеет большее разрешение, чем входное (рисунок 14).

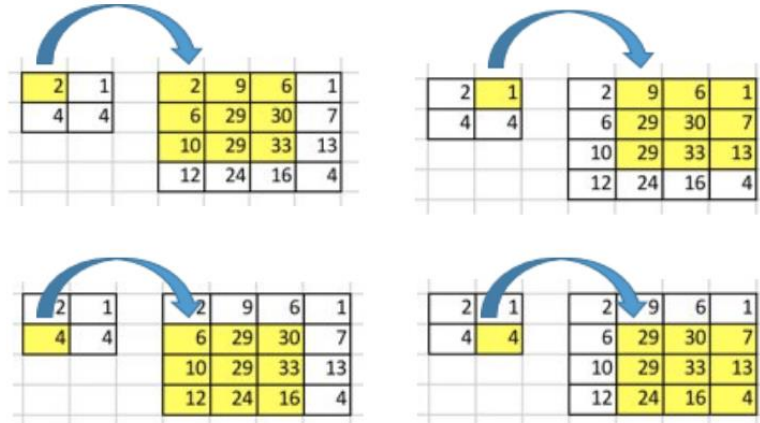


Рисунок 14. Транспонированная свертка

Отличие транспонированной свертки (рисунок 15) от обычной свертки заключается в том, что перед операцией свертки, между первоначальными элементами матрицы добавляются нулевые элементы. Это позволяет увеличить размер результирующей матрицы.

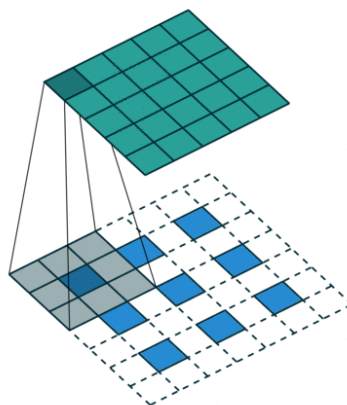


Рисунок 15. Операция транспонированной свертки

Сверточные слои извлекают признаки из изображений, но не уменьшают его размер, так как края изображения заполняются повторением крайних элементов.

Используемая функция активации во всех слоях, кроме последнего – Elu (рисунок 16).

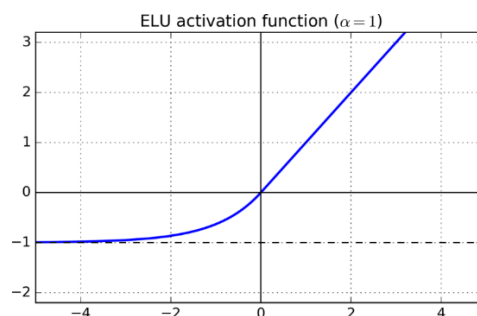


Рисунок 16. Активационная функция Elu

Функция активации, используемая на последнем слое – логистическая (сигмоидальная) функция (рисунок 17).

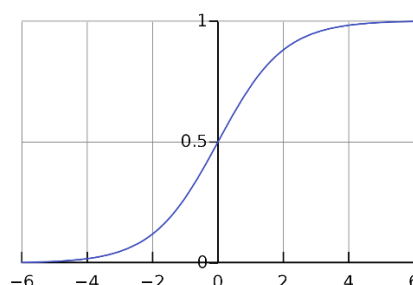


Рисунок 17. Сигмоида

3.4 Процесс обучения нейронной сети

Генерация выборки выполняется в программе Blender, куда загружается специальный скрипт, написанный на Python. Генерируется 10000 изображений.

Процесс обучения нейронной сети состоит из следующих шагов:

1. Считываются сгенерированные стереоизображения и карты глубины
2. Так как изображения в PNG и имеют 4 канала вместо 3, они преобразуются к RGB виду.

3. Карты глубины из трехканального изображения преобразуются в одноканальное, так как они черно-белые.
4. Каждое число в выборке делится на 255, так как на выходе нейронной сети данной структуры
5. Выборка случайно разбивается на тестовую, валидационную и обучающую в пропорции 10%/10%/80%.
6. Создается модель нейронной сети по приведенной выше структуре.
7. Обучающая выборка подается на вход модели.
8. Выполняется обучение модели, в качестве оптимизируемой функции – среднеквадратическая ошибка между истинными картами глубины и прогнозируемыми ответами. Модель обучается до тех пор, пока за последние 5 эпох есть уменьшение минимальной среднеквадратической ошибки на валидационной выборке (проверка выполняется каждую эпоху обучения).

3.5 Блок-схема алгоритма обучения нейронной сети

Блок-схема всей системы представлена в приложении.

4. Результаты проведенного исследования

По итогам обучения нейронной сети была оценена среднеквадратическая ошибка на тестовой выборке (рисунок 18), ошибка составила 0.0005.

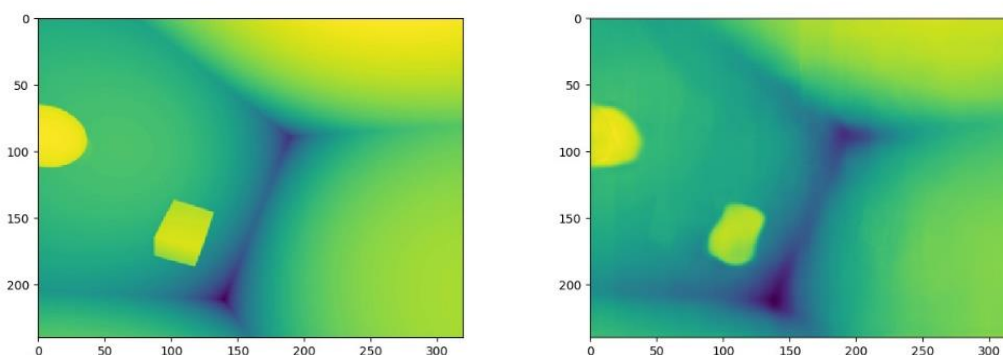


Рисунок 18. Истинные карты глубины (слева) и построенные с помощью нейронной сети (справа)

Обученную нейронную сеть можно загружать с помощью фреймворка Keras в другую программу и использовать для построения карт глубины. Несмотря на то, что используемая выборка была достаточно «синтетической», созданная структура нейронной сети может быть использована и с любой другой выборкой стереоизображений.

Для оценки актуальности созданной системы было проведено сравнение по качеству и производительности нейронной сети и метода построения карты глубины путем создания диспаратного изображения. Алгоритм создания диспаратного изображения был взят из библиотеки OpenCV. Результаты представлены в таблице 2.

Таблица 2. Сравнение нейронной сети и традиционного метода.

Метод	Среднеквадратическая ошибка	Время построения одного изображения, с	Частота обработки кадров, кадров/с
Нейронная сеть	0.005	0.020	50
OpenCV.StereoBM	0.09	0.006	167

Значение среднеквадратической ошибки в 0.09 является достаточно большим, если учесть, что пиксели кодированы вещественными числами в диапазоне от 0 до 1. Корень из среднеквадратической ошибки составляет 0.3, что характеризует метод библиотеки OpenCV как достаточно неточный. Кроме того, он требует подбора нескольких параметров (без подбора параметров, среднеквадратическая ошибка равна 0.24). Для подбора параметров была использована специальная программа с графическим интерфейсом (рисунок 19), созданная по причине трудоемкости подбора параметров для данного метода построения диспаратного изображения.

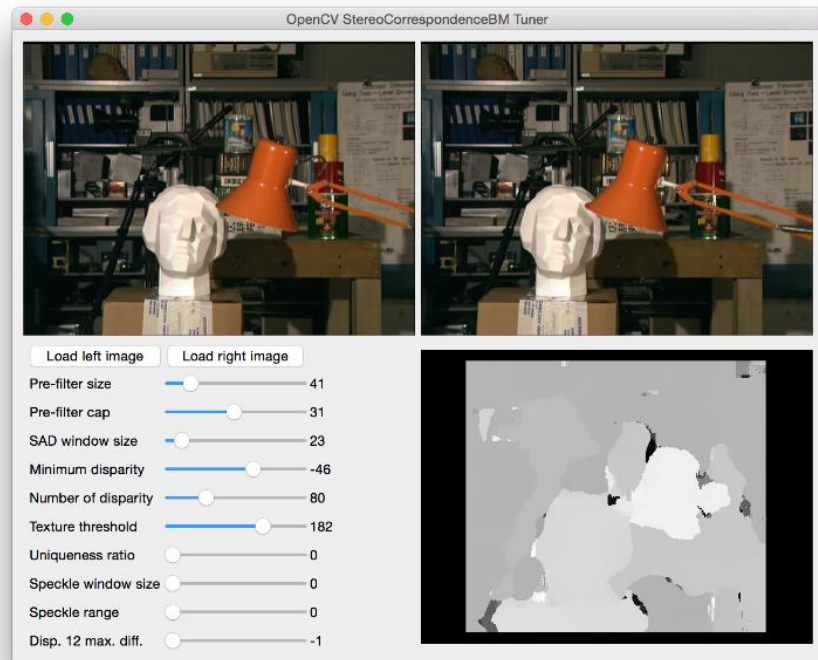


Рисунок 19. Программа, использовавшаяся для определения оптимальных параметров метода OpenCV

По проведенному выше сравнению можно сделать вывод, что разработка является перспективной.

5. Финансовый менеджмент, ресурсоэффективность и ресурсосбережение

ЗАДАНИЕ ДЛЯ РАЗДЕЛА «ФИНАНСОВЫЙ МЕНЕДЖМЕНТ, РЕСУРСОЭФФЕКТИВНОСТЬ И РЕСУРСОСБЕРЕЖЕНИЕ»

Студенту:

Группа	ФИО
8E41	Чугунов Роман Анварович

Институт		Кафедра	
Уровень образования	Бакалавриат	Направление/специальность	15.03.06 Мехатроника и робототехника

Исходные данные к разделу «Финансовый менеджмент, ресурсоэффективность и ресурсосбережение»:

1. Стоимость ресурсов научного исследования (НИИ): материально-технических, энергетических, финансовых, информационных и человеческих	- Оклад студента – 2410 руб. в месяц; - Оклад руководителя проекта – 18503 руб. в месяц. - Человеческие ресурсы – 2 человека (руководитель и студент-дипломник).
2. Нормы и нормативы расходования ресурсов	- Тариф на электроэнергию – 5,26 руб./кВт·ч.; - Годовая норма амортизации составляет 40 %
3. Используемая система налогообложения, ставки налогов, отчислений, дисконтирования и кредитования	На основании пункта 1 ст.58 закона №212-ФЗ для учреждений осуществляющих образовательную и научную деятельность вводится пониженная ставка – 27,1%

Перечень вопросов, подлежащих исследованию, проектированию и разработке:

1. Оценка коммерческого потенциала, перспективности и альтернатив проведения НИИ с позиции ресурсоэффективности и ресурсосбережения	- Методы коммерциализации результатов инженерных решений; - Морфологический анализ проекта.
2. Планирование и формирование бюджета научных исследований	- Определение трудоемкости выполнения работ; - Расчет материальных затрат НИИ; - Основная и дополнительная зарплата исполнителей темы; - Отчисления во внебюджетные фонды; - Накладные расходы;

	- Проведение анализа безубыточности проекта
3. <i>Определение ресурсной (ресурсосберегающей), финансовой, бюджетной, социальной и экономической эффективности исследования</i>	- Расчет интегрального показателя финансовой эффективности.

Перечень графического материала (с точным указанием обязательных чертежей):

1. *Оценка конкурентоспособности технических решений*
2. *Матрица SWOT*
3. *Альтернативы проведения НИ*
4. *График проведения и бюджет НИ*
5. *Оценка ресурсной, финансовой и экономической эффективности НИ*

Дата выдачи задания для раздела по линейному графику	01.03.2018
---	------------

Задание выдал консультант:

Должность	ФИО	Ученая степень, звание	Подпись	Дата
Доцент ОСГН	Петухов О.Н.	к.э.н.		01.03.2018

Задание принял к исполнению студент:

Группа	ФИО	Подпись	Дата
8Е41	Чугунов Роман Анварович		01.03.2018

5.1 Оценка коммерческого потенциала и перспективности проведения научных исследований с позиции ресурсоэффективности и ресурсосбережения.

Цель экономического раздела - провести детальный анализ проекта по критериям конкурентоспособности и ресурсоэффективности. Оценить перспективность проекта, определить трудоемкость и график работ, а также рассчитать интегральный показатель ресурсоэффективности.

5.1.1 Потенциальные потребители результатов исследования

В ходе выпускной квалификационной работы были проведены исследования потребности рынка в алгоритме построения карты глубины с помощью нейронных сетей.

Для определения потенциальных потребителей требуется определить целевой рынок и произвести его сегментирование. В данном случае целевым рынком являются:

- разработчики роботов, использующие стереопары в своих разработках;
- 3D-дизайнеры;

Конечными потребителями являются программисты-робототехники и дизайнеры.

5.1.2 Анализ конкурентных технических решений

Карты глубины используются в робототехнике для навигации и ориентации роботов со стереопарами. Также они могут использоваться в 3D дизайне для создания различных эффектов (например, тумана).



Рисунок 20. Карта глубины

Алгоритмы создания карт глубины являются совокупностью различных методов. Такое ПО является узкоспециализированным, известна только одна коммерчески продающаяся программа – StereoTracer компании Triaxes. Данное ПО позиционируется как инструмент для рекламных, полиграфических кампаний, фотостудий и фотографов. Программа Triaxes StereoTracer позволяет без специальной стереосъемки получать многоракурсные серии кадров и значительно ускорить работу по подготовке и печати лентикулярных 3D изображений. StereoTracer моделирует многоракурсную 3D съемку на основе одного исходного кадра или двух кадров стереопары. Программа реализует метод генерации кадров с использованием карты глубины. Полученная серия кадров экспортируется в программу 3DMasterKit для создания лентикулярного 3D изображения. Данная компания продает три версии StereoTracer с целью сегментировать рынок. Они отличаются набором функций, цены разнятся от 2125 руб. до 8840 руб.

Недостатком является то, что данное ПО не предназначено для использования в робототехнике, его использование в соответствующих вычислительных системах проблематично.

Из этого можно сделать вывод, что требуется разработать алгоритм, который будет доступен для использования как широкому кругу пользователей. К такому же выводу можно прийти проанализировав карту сегментации рынка (Таблица 3).

Таблица 3. Карта сегментации рынка

		Способ использования	
		Приложение с графическим интерфейсом пользователя	Алгоритм
	Дизайнеры	+	-

Сегменты рынка	Робототехники	-	+
----------------	---------------	---	---

Проведем анализ конкурентного технического решения с помощью оценочной карты.

Позиция разработки и конкурентов оценивается по каждому показателю экспертным путем по пятибалльной шкале, где 1 – наиболее слабая позиция, а 5 – наиболее сильная. Веса показателей, определяемые экспертным путем, в сумме должны составлять 1.

Таблица 4. Оценочная карта

Критерии оценки	Вес критерия	Баллы		Конкурентоспособность	
		Б _ф	Б _{к1}	К _ф	К _{к1}
1	2	3	4	5	6
Технические критерии оценки ресурсоэффективности					
1. Энергоэффективность	0.12	4	4	0.48	0.48
2. Удобство в эксплуатации	0.12	4	5	0.48	0.6
3. Помехоустойчивость	0.08	5	4	0.4	0.32
4. Ремонтпригодность	0.05	5	4	0.25	0.2
5. Надежность	0.05	5	4	0.25	0.2
6. Мобильность	0	5	5	0	0
7. Безопасность	0.12	4	4	0.48	0.48
8. Персонализация	0.12	5	4	0.6	0.48
9. Функциональная мощность	0.12	4	5	0.48	0.6
10. Качество интеллектуального интерфейса	0.10	4	4	0.4	0.4
Экономические критерии оценки ресурсоэффективности					
11. Цена	0.12	5	3	0.6	0.36
Итого	1	50	46	4.42	4.12

Где B_{ϕ} - разрабатываемый алгоритм, B_{kl} - программа компании Triaxes.
Конкурентоспособность рассчитываем по формуле:

$$K = \sum B_i * B_i$$

где K – конкурентоспособность научной разработки, V_i – вес показателя (в долях единицы), B_i – балл i -го показателя.

Преимущество перед конкурентом: низкая цена, помехоустойчивость, надежность ПО, ремонтпригодность (применительно к программному продукту), персонализация. Конкуренты выигрывают в удобстве эксплуатации, и функциональной мощности продукта.

Для улучшения разрабатываемого продукта применены другие технологии для достижения сходной цели.

5.1.3 Технология QuaD

Технология QuaD (QUality ADvisor) это инструмент измерения характеристик, который описывает качество новой разработки, а также ее перспективность на рынке. Технология позволяет принимать решение о целесообразности вложения капитала в НИР. Технология может использоваться при проведении различных маркетинговых исследований, существенно снижая их трудоемкость и повышая точность и достоверность результатов. Оценочная карта представлена в таблице 5.

Таблица 5. Таблица QuaD

Критерии оценки	Вес критерия	Баллы	Максимальный балл	Относительное значение (3/4)	Средневзвешенное значение (5x2)
1	2	3	4	5	6
Показатели оценки качества разработки					
1. Энергоэффективность	0.12	80	100	0.80	0.096

2. Помехоустойчивость	0.12	80	100	0.80	0.096
3. Надежность	0.08	100	100	1.00	0.08
4. Мобильность	0.05	100	100	1.00	0.05
5. Персонализация	0.05	100	100	1.00	0.05
6. Безопасность	0.00	100	100	1.00	0.00
7. Функциональная мощность (предоставляемые возможности)	0.12	80	100	0.80	0.096
8. Простота эксплуатации	0.12	100	100	1.00	0.12
9. Качество интеллектуального интерфейса	0.12	80	100	0.80	0.096
10. Ремонтопригодность	0.10	80	100	0.80	0.08
Показатели оценки коммерческого потенциала разработки					
11. Перспективность рынка	0.05	60	100	0.60	0.03
12. Цена	0,07	100	100	1.00	0.07
Итого	1			10.6	0,864

Оценка качества и перспективности по технологии QuaD определяется по формуле:

$$P_{cp} = \sum P_i * 100 = 0.864 * 100 = 86,4$$

где:

P_{cp} – средневзвешенное значение показателей качества и перспективности научной разработки;

P – средневзвешенное значение показателя.

Значение P_{cp} позволяет говорить о перспективах разработки и качестве проведенного исследования. Если значение показателя P_{cp} получилось от 100 до 80, то такая разработка считается перспективной. Если от 79 до 60 – то перспективность выше среднего. Если от 69 до 40 – то перспективность средняя. Если от 39 до 20 – то перспективность ниже среднего. Если 19 и ниже – то перспективность крайне низкая.

Из таблицы можно сделать вывод, что разработку можно считать перспективной.

5.1.4 SWOT-анализ

SWOT – Strengths (сильные стороны), Weaknesses (слабые стороны), Opportunities (возможности) и Threats (угрозы) – это комплексный анализ научно-исследовательского проекта. Такой анализ применяют для исследования внешней и внутренней среды проекта. Составим матрицу SWOT:

Таблица 6

	Сильные стороны научно-исследовательского проекта: С1. Надежность С2. Мобильность С3. Персонализация С4. Безопасность С5. Простота эксплуатации	Слабые стороны научно-исследовательского проекта: Сл1. Отсутствие прототипа научной разработки Сл2. Отсутствие финансирования Сл3. Повышенные требования к производительности «железа» Сл4. Отсутствие репутации на рынке
Возможности: В1. Использование инновационной инфраструктуры ТПУ В2. Повышение качества построения карт глубины В3. Появление дополнительного спроса на новый продукт	При использовании инфраструктуры ТПУ можно снизить стоимость проекта. Использование новых подходов (глубокое обучение) может повысить качество построения карт глубины.	Появление дополнительного спроса на рынке может способствовать финансированию проекта. Проводя исследования на базе ТПУ можно разработать прототип и получить известность на рынке

Угрозы:	Высокие требования к «железу» обусловлены спецификой разрабатываемого алгоритма. Качество работы можно повысить за счет сбора качественной выборки.	Необходимо разработать прототип алгоритма и пройти сертификацию для того, чтобы выйти на рынок и заполнить репутацию.
У1. Отсутствие спроса на разработку		
У2. Высокие требования к «железу» разработанного продукта		
У3. Неудовлетворительное качество работы разработки		

5.1.5 Морфологический анализ

Морфологический подход основан на систематическом исследовании всех теоретически возможных вариантов, которые вытекают из закономерностей объекта исследования. Анализ охватывает все возможные варианты. Путем комбинирования вариантов можно получить большое количество различных решений, ряд которых представляет практический интерес. Составим таблицу, в которой будут отражены возможные варианты исполнения по различным проблемам разработки.

Таблица 7. Морфологический анализ

Характеристика	Варианты исполнения		
	1	2	3
Язык программирования	C++	R	Python
Вид нейронных сетей	Сверточные	Капсюльные	Полносвязные
Реализация	Веб-приложение	Десктоп-приложение	Библиотека

На данный момент на рынке нет программы построения карт глубины достаточного качества и удобной в использовании для робототехников.

Использование нейронных сетей позволяет добиться нового уровня качества, однако они не используются в коммерческих решениях на данный момент.

5.2. Планирование научно-исследовательских работ

5.2.1. Структура работ в рамках научного исследования

Трудоемкость выполнения проекта оценивается в человеко-часах и зависит от множества факторов, которые сложно учесть при разработке. Для реализации проекта необходимо 2 исполнителя – научный руководитель (НР), студент (С). Этапы работы проекта представлены в таблице 8.

Таблица 8. Этапы работы проекта

Основные этапы	№ этапа	Содержание работ	Исполнитель
Разработка задания	1	Постановка задачи	НР, С
Выбор направления исследования	2	Обзор научно-технической базы	НР, С
	3	Разработка и утверждение ТЗ	НР, С
	4	Составление календаря проекта	С
	5	Разработка вариантов исполнения проекта	НР, С
Теоретические исследования	6	Сбор обучающих данных	С
	7	Разработка модуля чтения и предобработки данных	С
	8	Создание модели машинного обучения	С
	9	Обучение и настройка гиперпараметров алгоритма с учетом метрик качества	С
Экспериментальные исследования	10	Проверка работы алгоритма на тестовых данных	НР, С

Оформление отчета по НИР	11	Составление пояснительной записки	С
--------------------------	----	-----------------------------------	---

5.2.2 Определение трудоемкости выполнения работ

Основная часть стоимости разработки зачастую приходится на заработную плату исполнителей, поэтому важно определить трудоемкость каждого из участников. Ожидаемая трудоемкость находится по формуле

$$t_{ожi} = \frac{3 * t_{минi} + 2 * t_{маxi}}{5}$$

где $t_{ожi}$ – ожидаемая трудоемкость выполнения i -ой работы чел.-дн.;

$t_{минi}$ – минимально возможная трудоемкость выполнения заданной i -ой работы (оптимистическая оценка: в предположении наиболее благоприятного стечения обстоятельств), чел.-дн.;

$t_{маxi}$ – максимально возможная трудоемкость выполнения заданной i -ой работы (пессимистическая оценка: в предположении наиболее неблагоприятного стечения обстоятельств), чел.-дн.

Исходя из ожидаемой трудоемкости работ, определяется продолжительность каждой работы в рабочих днях T_{pi} , учитывающая параллельность выполнения работ несколькими исполнителями:

$$t_{pi} = \frac{t_{ожi}}{Ч_i}$$

где t_{pi} – продолжительность одной работы, раб.дн., $t_{ожi}$ – ожидаемая трудоемкость выполнения одной работы, чел.-дн, $Ч_i$ – численность исполнителей, выполняющих одновременно одну и ту же работу на данном этапе, чел.

Длительность каждого из этапов работ из рабочих дней следует перевести в календарные дни. Для этого необходимо воспользоваться следующей формулой. Для примера произведём расчёт первого этапа работы руководителя:

$$T_{ki} = T_{pi} * k_{кал}$$

где t_{ki} – продолжительность выполнения i -й работы в календарных днях; t_{pi} – продолжительность выполнения i -й работы в рабочих днях;
 $k_{\text{кал}}$ – коэффициент календарности.

Коэффициент календарности определяется по следующей формуле:

$$k_{\text{кал}} = \frac{t_{\text{кал}}}{t_{\text{кал}} - t_{\text{вых}} - t_{\text{пр}}} = \frac{365}{365 - 104 - 16} = 1,5$$

где: $t_{\text{кал}}$ – количество календарных дней в году;

$t_{\text{вых}}$ – количество выходных дней в году;

$t_{\text{пр}}$ – количество праздничных дней в году.

В таблице находятся расчеты этапов отдельных видов работ.

Таблица 9. Временные показатели проведения научного исследования

Название работы	Трудоемкость работ						Длительно сть работ в рабочих днях		Длительнос ть работ в календарны х днях	
	t_{min} чел-дни		t_{max} чел-дни		$t_{\text{ож}}$ чел-дни		t_{pi}		t_{ki}	
	Студент	Научный руководитель	Студент	Научный руководитель	Студент	Научный руководитель	Одновременн о выполнени е работ		Одновременное выполнение работ	
							Студент	Научный руководит	Студент	Научный руководит
Постановка задачи	5	3	8	6	6,2	4,2	3,1	2,1	5	4
Обзор научно-технической базы	7	2	12	4	9	2,8	4,5	1,4	7	3
Разработка и утверждение ТЗ	7	1	12	2	9	1,4	4,5	0,7	7	2
Составление календаря проекта	3	0	5	0	3,8	0	3,8	0	6	0

Разработка вариантов исполнения проекта	9	4	16	7	11,8	5,2	5,9	2,6	9	4
Сбор обучающих данных	5	0	9	0	6,6	0	6,6	0	10	0
Разработка модуля чтения и предобработки данных	10	0	20	0	14	0	14	0	21	0
Создание модели машинного обучения	10	0	20	0	14	0	14	0	21	0
Обучение и настройка гиперпараметров алгоритма с учетом метрик качества	10	0	20	0	14	0	14	0	21	0
Проверка работы алгоритма на тестовых данных	6	1	14	8	9,2	3,8	4,6	1,9	7	3
Составление пояснительной записки	7	0	14	0	9,8	0	9,8	0	15	0
Итого									129	16

5.2.3 Разработка графика проведения научного исследования

По таблице 9 составим таблицу 10 – продолжительность работ проведения НИОКР.

Таблица 10. Продолжительность работ проведения НИОКР

№ этапа	Этап	Исполнитель	T_{ki}
1	Постановка задачи	НР	4
		С	5
2	Обзор научно-технической базы	НР	3
		С	7
3	Разработка и утверждение ТЗ	НР	2
		С	7
4	Составление календаря проекта	С	6
5	Разработка вариантов исполнения проекта	НР	4
		С	9
6	Сбор обучающих данных	С	10
7	Разработка модуля чтения и предобработки данных	С	21
8	Создание модели машинного обучения	С	21
9	Обучение и настройка гиперпараметров алгоритма с учетом метрик качества	С	21
10	Проверка работы алгоритма на тестовых данных	НР	3
		С	7
11	Составление пояснительной записки	С	15

По данным из таблицы 8 «Продолжительность работ проведения НИОКР» создадим диаграмму Ганта студента (таблица 11) и преподавателя (таблица 12), которая строилась при максимальном количестве дней каждой работы.

Таблица 11. Диаграмма Ганта. График работы студента.

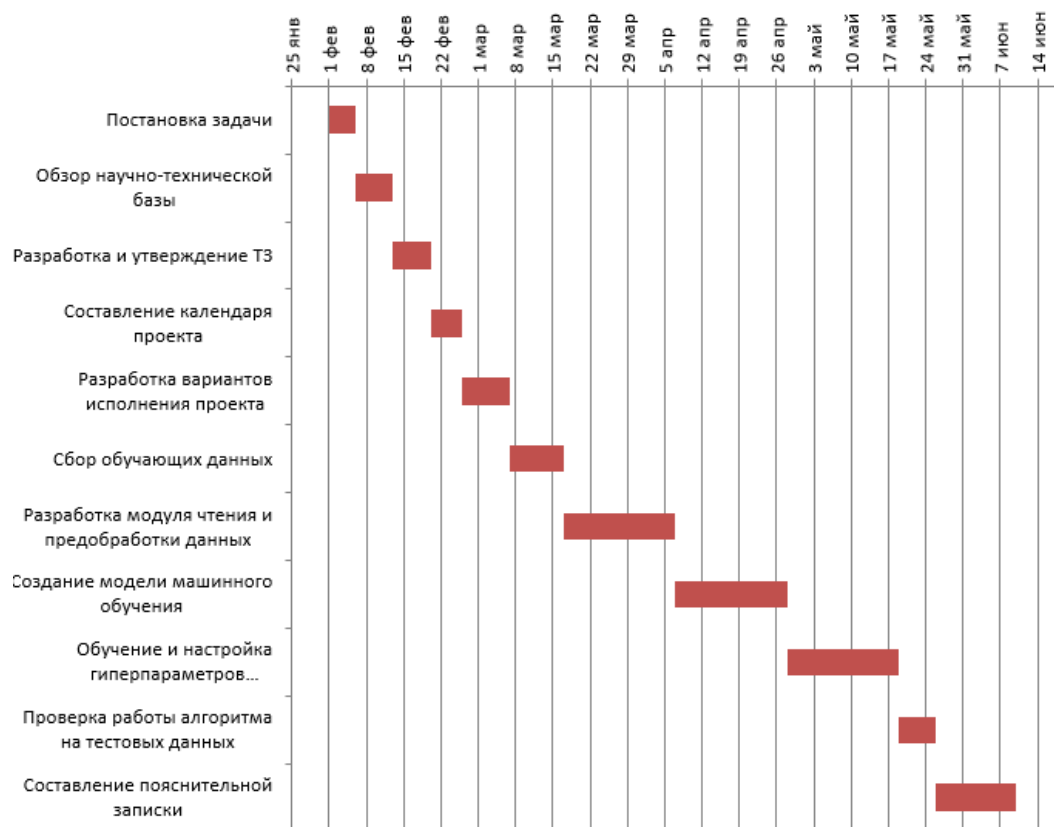
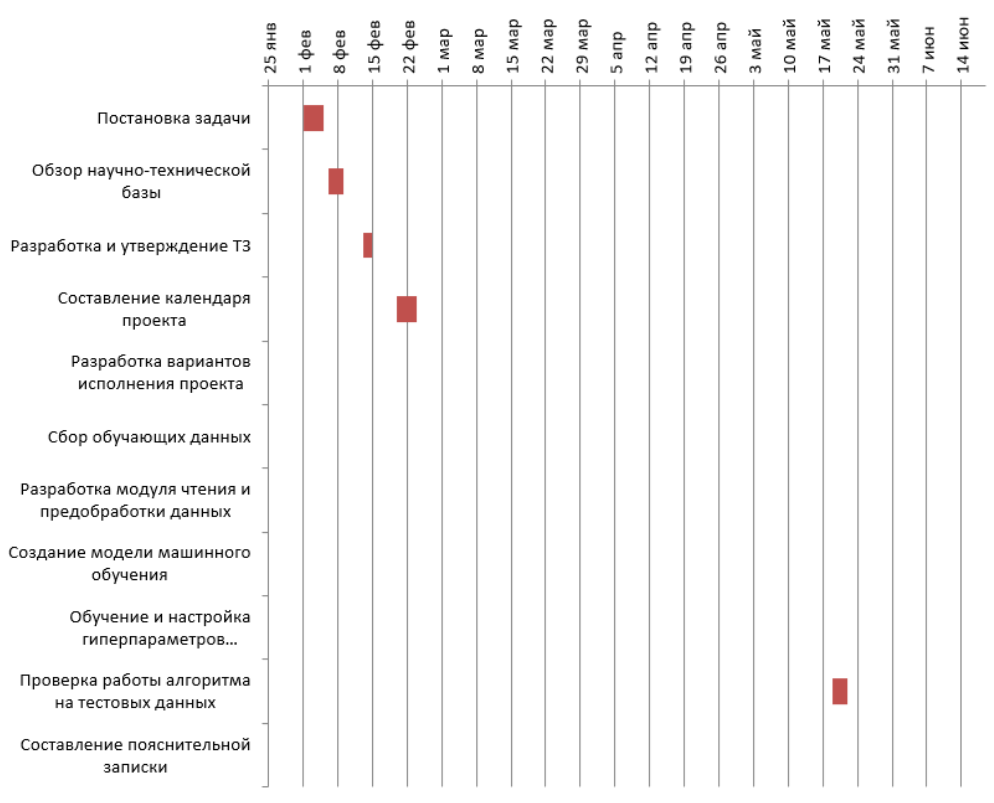


Таблица 12. Диаграмма Ганта. График работы научного руководителя



5.2.4 Бюджет научно-технического исследования

Бюджет научно-технического исследования должен быть основан на достоверном отображении всех видов расходов, связанных выполнением проекта. В процессе формирования бюджета разработки используется следующая группировка затрат по статьям:

- материальные затраты разработки;
- основная заработная плата исполнителей темы;
- дополнительная заработная плата исполнителей темы;
- отчисления во внебюджетные фонды (страховые отчисления);
- затраты на научные и производственные командировки;
- накладные расходы.

5.2.5 Расчет материальных затрат

Расчет материальных затрат проводить не требуется, так как они отсутствуют. Используется материально-техническая база ТПУ.

5.2.6 Основная заработная плата исполнителям темы

Статья включает основную заработную плату работников, непосредственно занятых выполнением НТИ, (включая премии, доплаты) и дополнительную заработную плату. Она рассчитывается по формуле

$$З_{зп} = З_{осн} + З_{доп}$$

где $З_{осн}$ – основная заработная плата;

$З_{доп}$ – дополнительная заработная плата (12-20 % от $З_{осн}$).

Среднедневная заработная плата рассчитывается по формуле:

$$З_{дн} = \frac{З_{м} * М}{F_{д}}$$

где $З_{м}$ – месячный должностной оклад работника, руб.;

$М$ – количество месяцев работы без отпуска в течение года: при отпуске в 24 раб. дня $М=11,2$ месяца, 5-дневная неделя; при отпуске в 48 раб. дней $М=10,4$ месяца, 6-дневная неделя; при отпуске в 72 раб. дней $М=9,6$.

F_d – действительный годовой фонд рабочего времени научно- технического персонала, раб. дн.

Таблица 13. Баланс рабочего времени

Показатели рабочего времени	Руководитель	Студент
Календарное число дней	365	365
Количество нерабочих дней - выходные дни - праздничные дни	120	120
Потери рабочего времени - отпуск - невыходы по болезни	48	72
Действительный годовой фонд рабочего времени	197	173

Месячный оклад работника рассчитывается по формуле:

$$Z_m = Z_{тс} * (1 + k_{пр} + k_d) * k_p$$

где $Z_{тс}$ – заработная плата по тарифной ставке, руб.;

$k_{пр}$ – премиальный коэффициент, равный 0,3 (т.е. 30% от $Z_{тс}$);

k_d – коэффициент доплат и надбавок составляет примерно 0,2 – 0,5 (в НИИ и на промышленных предприятиях – за расширение сфер обслуживания, за профессиональное мастерство, за вредные условия: 15-20% от $Z_{тс}$);

k_p – районный коэффициент, равный 1,3 (для Томска).

Расчёт основной заработной платы приведён в таблице 14

Таблица 14. Расчет основной заработной платы

Исполнители	Разряд	$Z_{тс}$,	$k_{пр}$	k_d	k_p	Z_m ,	$Z_{дн}$,	T_p ,	$Z_{осн}$,
-------------	--------	------------	----------	-------	-------	---------	------------	---------	-------------

		руб.				руб	руб.	раб. дн.	руб.
Руководитель	1	9489	0,3	0,2	1,3	18503,55	976,83	16	15629,28
Студент		18540	0	0	1,3	2410,2	130,95	129	16829,55
Итого $Z_{\text{осн}}$									=32458,83

5.2.7 Дополнительная заработная плата

Дополнительная заработная плата включает заработную плату за не отработанное рабочее время, но гарантированную действующим законодательством. Расчет дополнительной заработной платы ведется по формуле:

$$Z_{\text{доп}} = k_{\text{доп}} * Z_{\text{осн}}$$

где $k_{\text{доп}}$ – коэффициент дополнительной заработной платы (на стадии проектирования принимается равным 0,12 – 0,15).

$k_{\text{доп}}$ равен 0,12. Результаты по расчетам дополнительной заработной платы сведены в таблицу 15.

Таблица 15. Затраты на дополнительную заработную плату

Исполнители	Основная зарплата (руб.)	Коэффициента дополнительной заработной платы ($k_{\text{доп}}$)	Дополнительная зарплата (руб.)
Руководитель	15629,28	0,12	1875,51
Студент	16829,55	0,12	2019,55
Итого:			3895,06

5.2.8 Отчисления во внебюджетные фонды (страховые отчисления)

Величина отчислений во внебюджетные фонды определяется исходя из формулы:

$$Z_{\text{внеб}} = k_{\text{внеб}} * (Z_{\text{осн}} + Z_{\text{доп}})$$

где $k_{\text{внеб}}$ – коэффициент отчислений на уплату во внебюджетные фонды (пенсионный фонд, фонд обязательного медицинского страхования и пр.).

На 2018 г. в соответствии с Федеральным законом от 24.07.2009 №212-ФЗ установлен размер страховых взносов равный 30%. На основании пункта 1 ст.58 закона №212-ФЗ для учреждений, осуществляющих образовательную и научную деятельность в 2018 году пониженная ставка – 27,1%. Отчисления во внебюджетные фонды представлены в таблице 16.

Таблица 16. Отчисления во внебюджетные фонды

Исполнитель	Основная заработная плата, руб.	Дополнительная заработная плата, руб.
Руководитель проекта	15629,28	1875,51
Студент	16829,55	2019,55
Коэффициент отчислений во внебюджетные фонды	27,1%	
Итого		
Руководитель	4743,80	
Студент	5108,11	
Итого	9851,91	

5.2.9 Накладные расходы

Накладные расходы учитывают прочие затраты организации, не попавшие в предыдущие статьи расходов: печать и ксерокопирование материалов, оплата услуг связи, электроэнергии и т.д. Расчет накладных расходов определяется по формуле:

$$Z_{\text{нак}} = \sum C_{\text{т}} * k_{\text{нр}}$$

где $k_{\text{нр}}$ – коэффициент, учитывающий накладные расходы,

$C_{\text{т}}$ – затраты по статьям накладных расходов

Величину коэффициента накладных расходов можно взять в размере 5%.

$$Z_{\text{нак}} = (9851,91 + 17504,79 + 18849,1) * 0.05 = 2310,29 \text{ руб.}$$

5.2.10 Контрагентные расходы

Контрагентные расходы включают затраты, связанные с выполнением каких-либо работ по теме сторонними организациями (контрагентами, субподрядчиками)

В данном проекте есть контрагентные работы отсутствуют.

5.2.11 Формирование бюджета затрат научно-исследовательского проекта

Рассчитанная величина затрат научно-исследовательской работы является основой для формирования бюджета затрат проекта. Определение бюджета затрат на научно-исследовательский проект по каждому варианту исполнения приведен в таблице 17.

Таблица 17. Расчет бюджета затрат НТИ

Наименование статьи	Сумма (руб.)
1. Материальные затраты НТИ	0
2. Затраты на заработную плату научному руководителю	17504,79
3. Затраты на заработную плату студенту	18849,1
4. Затраты на отчисления во внебюджетный фонд	9851,91
5. Накладные расходы	2310,29
Бюджет затрат НТИ	48516,09

ТПУ частично финансирует проект, предоставляя аудиторию для работы, ПК, заработную плату и иных ресурсов.

5.2.12 Определение ресурсной, финансовой и экономической эффективности ресурсов

В результате исследования были определены затраты на проект по разработке алгоритма построения карт глубины. Бюджет составляет 48516,09 рублей. Приняв во внимание все преимущества разрабатываемого ПО, можно предположить, что продукт будет иметь спрос на рынке и станет достойным аналогом своим конкурентам.

6. Социальная ответственность

ЗАДАНИЕ ДЛЯ РАЗДЕЛА «СОЦИАЛЬНАЯ ОТВЕТСТВЕННОСТЬ»

Студенту:

Группа	ФИО
8Е41	Чугунов Роман Анварович

Школа	ИШИТР	Отделение	ОАР
Уровень образования	Бакалавриат	Направление/специальность	15.03.06 Мехатроника и робототехника

Исходные данные к разделу «Социальная ответственность»:

1. Характеристика объекта исследования (вещество, материал, прибор, алгоритм, методика, рабочая зона) и области его применения

Объектом исследования является алгоритм построения карт глубины изображения по стереоизображению с применением методов машинного обучения. Оборудованием, на котором производится работа, является персональный компьютер.

Разработанная система может использоваться как в области создания компьютерной 3D графики (в частности, создании эффектов тумана или дыма), рендеринге 3D сцен, так и в робототехнике: карты глубины используются в робототехнике для навигации и ориентации роботов.

Перечень вопросов, подлежащих исследованию, проектированию и разработке:

<p>1. Производственная безопасность</p>	<p>Анализ выявленных вредных факторов:</p> <ul style="list-style-type: none"> - Освещение - Микроклимат - Шум - Психофизиологические факторы: нервно-психические перегрузки <p>Анализ выявленных опасных факторов:</p> <ul style="list-style-type: none"> - Электрический ток (источник – ПК) - Короткое замыкание - Статическое электричество (источник – ПК)
<p>2. Экологическая безопасность:</p>	<p>Воздействие объекта на атмосферу, гидросферу отсутствует.</p> <p>Воздействие на литосферу происходит при утилизации ПК, используемого для разработки, а также утилизации люминесцентных ламп освещения.</p>
<p>3. Безопасность в чрезвычайных ситуациях:</p>	<p>Возможной чрезвычайной ситуацией при разработке алгоритма является возникновение пожара на рабочем месте.</p>
<p>4. Правовые и организационные вопросы обеспечения безопасности:</p>	<p>- ГОСТ 12.0.003-74 «Опасные и вредные производственные факторы. Классификация».</p>

	<ul style="list-style-type: none"> - СанПиН 2.2.2/2.4.1340-03. - СанПиН 2.2.4.548-96 - ТОИ Р-45-084-01 - «Правила устройства электроустановок» - «Правила технической эксплуатации электроустановок потребителей» - «Межотраслевые правила охраны труда (правила безопасности) при эксплуатации электроустановок» - ГОСТ Р 53692-2009. «Ресурсосбережение. Обращение с отходами.» - Федеральный закон № 89 от 1998г. «Об отходах производства и потребления» - Постановление Правительства РФ № 340 от 2002 г. - ГОСТ 12.1.033-81 Система стандартов безопасности труда (ССБТ). Пожарная безопасность. Термины и определения. - Трудовой Кодекс РФ
--	--

Дата выдачи задания для раздела по линейному графику	01.03.2018
---	------------

Задание выдал консультант:

Должность	ФИО	Ученая степень, звание	Подпись	Дата
ассистент	Авдеева Ирина Ивановна			01.03.2018

Задание принял к исполнению студент:

Группа	ФИО	Подпись	Дата
8Е41	Чугунов Роман Анварович		01.03.2018

Введение

Целью данной выпускной квалификационной работы является разработка алгоритма построения карт глубины по стереоизображениям с помощью методов машинного обучения.

Карта глубины — это изображение, которое хранит информацию о расстоянии от камеры до видимых объектов.

В данном разделе ВКР исследованы меры по защите работника от возможного негативного воздействия среды, а также вредные и опасные факторы среды. Кроме того, были рассмотрены возможные чрезвычайные ситуации и действия, которые разработчик должен выполнить в случае возникновения чрезвычайной ситуации.

Разработанная система может использоваться как в области создания компьютерной 3D графики (в частности, создании эффектов тумана или дыма), рендеринге 3D сцен, так и в робототехнике: карты глубины используются в робототехнике для навигации и ориентации роботов.

Были выделены и рассмотрены такие факторы, воздействующие на разработчика алгоритма, как: освещение, микроклимат, электромагнитное излучение, нервно-психические перегрузки. К опасным факторам при работе с персональным компьютером можно отнести высокое напряжение и возможность короткого замыкания, влекущего за собой опасность поражения разработчика электрическим током. Рассматриваются вопросы правового регулирования трудовых отношений, связанных с использованием разработанной системы.

Программные продукты не оказывают непосредственного негативного влияния на окружающую среду, однако их использование сопряжено с использованием персональных компьютеров, что оказывает негативное влияние на литосферу при утилизации ПК.

6.1 Производственная безопасность

Вредные и опасные факторы, воздействующие на сотрудника, устанавливаются согласно ГОСТ 12.0.003-74 «Опасные и вредные производственные факторы. Классификация».

Опасные и вредные факторы при выполнении работ по разработке алгоритма построения карт глубины по стереоизображению:

Таблица 18. Вредные и опасные факторы

Источник фактора, наименование видов работ	Факторы (по ГОСТ 12.0.003-74)		Нормативные документы
	Вредные	Опасные	
Разработка алгоритма построения карт глубины по стереоизображению на ПК	<ol style="list-style-type: none"> 1. Отклонение показателей освещенности на рабочем месте. 2. Повышенный уровень шума на рабочем месте. 3. Отклонение показателей микроклимата на рабочем месте. 4. Психофизиологические факторы: нервно-психические перегрузки. 	<ol style="list-style-type: none"> 1. Электрический ток (причиной является использование ПК). 2. Возможность короткого замыкания. 3. Повышенный уровень статического электричества. 	<p>Освещение, шум, статическое электричество, психофизиологические факторы:</p> <p>СанПин 2.2.2/2.4.1340-03 [1]</p> <p>ТОИ Р-45-084-01[2]</p> <p>Микроклимат:</p> <p>СанПин 2.2.4.548-96 [3]</p> <p>СанПин 2.2.2/2.4.1340-03</p> <p>Электрический ток, короткое замыкание:</p>

			<p>«Правила устройства электроустановок» утвержденные Главтехуправлением, Госэнергонадзором Минэнерго СССР 05.10.1979 г. [4]</p> <p>ГОСТ 12.1.033-81 [5]</p> <p>«Правила технической эксплуатации электроустановок потребителей», утвержденные Приказом Минэнерго России от 13.01.2003 г. №6 [6]</p> <p>«Межотраслевые правила охраны труда (правила безопасности) при эксплуатации электроустановок» (ПОТ РМ 016-2001), утвержденные Постановлением Минтруда России от 05.01.2001 г. №3. [7]</p>
--	--	--	---

6.1.1 Освещение

В данном случае к негативным факторам относятся повышенные урони излучения, а также увеличенная нагрузка на зрительные органы.

Требования к освещению установлены в СанПиН 2.2.2/2.4.1340-03 (таблица 19).

Таблица 19. Требования к освещению на рабочих местах, оборудованных ПЭВМ

Освещенность на рабочем столе	300-500 лк
Освещенность на экране ПЭВМ	Не выше 300 лк
Блики на экране	Не выше 40 кд/м ²
Прямая блескость источника света	200 кд/м ²
Показатель ослепленности	Не более 20
Показатель дискомфорта	Не более 15
Отношение яркости между рабочими поверхностями	3:1-5:1 10:1
Коэффициент пульсации	Не более 5%

6.1.2 Шум

В производственных помещениях при выполнении основных или вспомогательных работ с использованием ПЭВМ уровни шума на рабочих местах не должны превышать предельно допустимых значений, установленных для данных видов работ в соответствии с действующими санитарно-эпидемиологическими нормативами.

В СанПиН 2.2.2/2.4.1340-03 установлены допустимые значения уровней звукового давления, создаваемого ПЭВМ (таблица 20).

Таблица 20. Допустимые значения уровней звукового давления в октавных полосах частот и уровня звука, создаваемого ПЭВМ

Уровни звукового давления в октавных полосах со среднегеометрическими частотами									Уровни звука в дБА
31,5 Гц	63 Гц	125 Гц	250 Гц	500 Гц	1000 Гц	2000 Гц	4000 Гц	8000 Гц	
86 дБ	71 дБ	61 дБ	54 дБ	49 дБ	45 дБ	42 дБ	40 дБ	38 дБ	50

6.1.3 Микроклимат

В производственных помещениях, в которых работа с использованием ПЭВМ является основной (диспетчерские, операторские, расчетные, кабины и посты управления, залы вычислительной техники и др.) и связана с нервно-эмоциональным напряжением, должны обеспечиваться оптимальные параметры микроклимата для категории работ 1а и 1б в соответствии с действующими санитарно-эпидемиологическими нормативами микроклимата производственных помещений. На других рабочих местах следует поддерживать параметры микроклимата на допустимом уровне, соответствующем требованиям указанных выше нормативов.

Содержание вредных химических веществ в производственных помещениях, в которых работа с использованием ПЭВМ является основной (диспетчерские, операторские, расчетные, кабины и посты управления, залы вычислительной техники и др.), не должно превышать предельно допустимых концентраций загрязняющих веществ в атмосферном воздухе населенных мест в соответствии с действующими гигиеническими нормативами.

Также указывается, что в помещениях с ПЭВМ должна ежедневно проводиться влажная уборка.

Уставленные гигиенические нормативы для помещений с ВДТ и ПЭВМ для категории работы 1б приведены в таблице 21.

Таблица 21. Оптимальные величины показателей микроклимата на рабочих местах производственных помещений (СанПиН 2.2.4.548-96).

Период года	Категория работ по уровню энергозатрат	Температура воздуха, °С	Температура поверхностей, °С	Относительная влажность воздуха, %	Скорость движения воздуха, м/с
Холодный	1б (140-174 Вт)	21-23	20-24	60-40	0.1
Теплый		22-24	21-25	60-40	0.1

6.1.4 Статическое электричество

В помещениях, оборудованных ПЭВМ, токи статического электричества чаще всего возникают при прикосновении персонала к любому из элементов ПЭВМ. Такие разряды опасности для человека не представляют, однако кроме неприятных ощущений могут привести к выходу оборудования из строя.

Для предотвращения образования и защиты от статического электричества в помещении используются нейтрализаторы и увлажнители, а полы имеют антистатическое покрытие в виде поливинилхлоридного антистатического линолеума.

Также в СанПиН 2.2.2/2.4.1340-03 установлен максимальный допустимый электростатический потенциал экрана видеомонитора – 500 В.

В качестве мер уменьшения влияния вредных факторов на пользователя используются защитные фильтры для мониторов, увлажнители воздуха. Должны использоваться розетки с заземлением. Требуется проводить регулярную влажную уборку.

6.1.5 Психофизиологические факторы

Работа с ПК сопряжена с воздействием вредных психофизиологических факторов, в частности, нервно-психических перегрузок. Для снижения воздействия вредных факторов, устанавливаются перерывы в работе для отдыха сотрудников. Суммарное время регламентированных перерывов при работе с ПК зависит от категории трудовой деятельности и уровня нагрузки за рабочую смену. В таблице 22 приведено суммарное время отдыха для каждой категории работ.

Таблица 22. Суммарное время перерывов в зависимости от категории работы и нагрузки

	Уровень нагрузки за рабочую смену при видах работ с ПЭВМ	Суммарное время регламентированных
--	--	------------------------------------

Категория работы с ПЭВМ	группа А, количество знаков	группа Б, количество знаков	группа В, часов	перерывов при 8-часовой смене, мин.
I	до 20 000	до 15 000	до 2	50
II	до 40 000	до 30 000	до 4	70
III	до 60 000	до 40 000	до 6	90

В данном случае уровень нагрузки относится к группе В, категория работы III. Согласно таблице, требуется установить перерывы, сумма которых за смену составит не менее 90 минут. По типовой инструкции по охране труда при работе на персональном компьютере ГОИ Р-45-084-01 для данной категории работ требуется установить перерывы по 15 минут каждый трудовой час.

6.1.6 Электрический ток

К опасностям использования электрического тока относятся возможность поражения электрическим током, а также воспламенения электронных устройств из-за воздействия различных условий – попадания влаги или нарушения изоляции

Поражение электрическим током может привести к ожогам, судорогам, повреждению нервной системы, а также смерти. Возникновение пожара может привести к последствиям, описанным в ГОСТ 12.1.033-81.

Во избежание смерти и других негативных эффектов необходимо соблюдать правил пожарной и электрической безопасности. Подготовка к возникновению данных ситуаций должна производиться до начала работы.

Требования безопасности при эксплуатации электрооборудования регламентируются следующими нормативными актами:

- Правилами устройства электроустановок (издание шестое с отдельными разделами и главами в издании седьмом), утвержденными Главтехуправлением, Госэнергонадзором Минэнерго СССР 05.10.1979 г.;

- Правилами технической эксплуатации электроустановок потребителей, утверждёнными Приказом Минэнерго России от 13.01.2003 г. №6;
- Межотраслевыми правилами охраны труда (правилами безопасности) при эксплуатации электроустановок (ПОТ РМ 016-2001), утвержденными Постановлением Минтруда России от 05.01.2001 г. №3.

Согласно им:

- электрооборудование, имеющее контакты для подключения заземления, должно быть заземлено, а помещения, где размещаются рабочие места с ПЭВМ, должны быть оборудованы защитным заземлением (занулением) в соответствии с техническими требованиями по эксплуатации оборудования;
- все крышки и защитные панели должны находиться на своих местах (при отсутствии крышки или защитной панели эксплуатация электрооборудования не допускается);
- при работе с электрооборудованием не допускать попадания влаги на поверхность электрооборудования, а также запрещается работать на электрооборудовании влажными руками;
- вентиляционные отверстия электрооборудования не должны быть перекрыты находящимися вплотную стенами, мебелью, посторонними предметами;
- выдергивание штепсельной вилки электроприбора необходимо осуществлять за корпус штепсельной вилки, при необходимости придерживая другой рукой корпус штепсельной розетки;
- подключение и отключение разъемов компьютеров и оргтехники должно производиться при отключенном питании (за исключением подключения и отключения USB-устройств);
- удаление пыли с электрооборудования должно производиться в отключенном от электрической цепи состоянии;

- перед использованием электроприборов необходимо проверить надёжность крепления электророзетки, свериться с номиналом используемого напряжения;
- корпуса штепсельных розеток и выключателей не должны содержать трещин, оплавлений и других дефектов, способных снизить защитные свойства или нарушить надёжность контакта;
- кабели (шнуры) электропитания не должны содержать повреждений изоляции, сильных изгибов и скручиваний;

6.1.7 Заключение по разделу

По итогам анализа было выявлено следующее:

- Освещение на рабочем месте соответствует нормам – используется несколько энергосберегающих ламп.
- Уровни шума находятся в допустимых пределах – источником шума при эксплуатации ПК могут являться системы охлаждения, а также жесткий диск, однако уровень создаваемого ими шума невысок.
- Микроклиматические условия соблюдаются за счет использования систем отопления и кондиционирования и соответствуют установленным нормам.
- Защита от повреждений электроники статическим электричеством не обеспечивается, однако так как корпус ПК закрыт, вероятность поражения элементов или работника минимальна (если не прикасаться мокрыми руками к корпусу).
- Во время работы делаются перерывы для снижения нагрузки и предотвращения нервно-психических перегрузок.
- Помещение оборудовано согласно требованиям электробезопасности.
- Регулярно проводится влажная уборка помещения.

6.2 Экологическая безопасность

Для разработки системы построения карт глубины необходим компьютер, следовательно, в нашем случае воздействие на литосферу происходит при утилизации персонального компьютера. ПК может нагреваться, а также быть источником электромагнитного и ионизирующего излучения, а также шума. Для защиты здоровья сотрудников, работающих с программным обеспечением, рекомендуется соблюдать необходимую дистанцию при работе с компьютером (от 0.5 до 1 м), а также использовать эффективные охлаждающие системы и наиболее современное и эффективное оборудование.

Федеральный закон № 89 от 1998г. «Об отходах производства и потребления» [7] запрещает юридическим лицам самовольно избавляться от опасных отходов. Этим видом деятельности, согласно постановлению Правительства РФ № 340 от 2002 г. [8], могут заниматься только специализированные структуры. В их число входят и фирмы, которые занимаются утилизацией электронных отходов.

Обращение с отходами регламентируется ГОСТ Р 53692-2009 «Ресурсосбережение. Обращение с отходами.» [9]

Поэтому, при необходимости утилизировать вышедшую из употребления электронику наиболее безопасным для окружающей среды способом необходимо обращаться в специализированную компанию по утилизации. Такие компании действуют на всей территории Российской Федерации, в том числе и в Томской области. Необходимо отметить, что в целом при работе с компьютером существенного загрязнения окружающей среды не происходит и вредные выбросы не сравнимы с производственными.

Также существуют компании, занимающиеся утилизацией энергосберегающих ламп. Причина опасности данных ламп заключается в наличии ртути в их составе. Специализированные компании занимаются демеркуризацией и утилизацией ртутных отходов. Данные компании имеют

специальные лицензии на сбор, использование, транспортировку отходов. В случае выхода из строя используемой электроники или ламп, отходы передаются в соответствующие компании.

6.3 Безопасность в чрезвычайных ситуациях

Наиболее вероятной чрезвычайной ситуацией при разработке алгоритма является пожар на рабочем месте. В качестве противопожарных мероприятий должны быть применены следующие меры:

- в помещении должны находиться средства тушения пожара, средства связи;
- электрическая проводка электрооборудования и осветительных приборов должна быть исправна;
- все сотрудники должны знать место нахождения средств пожаротушения и уметь ими воспользоваться, средств связи и номера экстренных служб.

В связи с возможностью возникновения пожара разработан следующий план действий:

- в случае возникновения пожара сообщить о нем руководителю, постараться устранить очаг возгорания имеющимися силами при помощи первичных средств пожаротушения (огнетушитель порошковый, углекислотный О-1ПО (з)-АВСЕ);
- привести в действие ручной пожарный извещатель, если очаг возгорания потушить не удастся;
- сообщить о возгорании в службу пожарной охраны по телефону 01 или 010, сообщить адрес, место и причину возникновения пожара;
- принять меры по эвакуации людей и материальных ценностей;
- встретить пожарную охрану, при необходимости сообщить всю необходимую информацию и оказать помощь при выборе наилучшего подхода к очагу возгорания.

Рабочее помещение оборудовано в соответствии с требованиями пожарной безопасности. Имеется порошковый огнетушитель, а также пожарная сигнализация и средства связи.

6.4 Правовые и организационные вопросы обеспечения безопасности

Работодатели при установке ПК обязаны выполнить следующий перечень требований:

- к помещению;
- к освещению;
- к организации медицинского обследования пользователей;
- и многое другое.

Немаловажным фактором при выборе компьютеров для сотрудников является возможность конструкции компьютера изменять положение ПК в различных плоскостях (горизонтальные или вертикальные), с возможной устойчивой фиксацией в положении, которая удобна пользователю. Цвет корпуса ПК должен быть нежным, спокойным без блестящих деталей, который создавая блики, вызывают повышенную утомляемость глаз. Экран монитора должен содержать регулировку яркость и контрастности, что каждый работник мог установить нужный режим, которые будет соответствовать чувствительности глаз.

Устройство рабочего стола должна быть использовано для оптимального размещения используемого оборудования. Кроме того, форма рабочего стола должна быть удобна для поддержания рациональной позы пользователя, так что бы он мог менять положения своего тела для предупреждения утомления. На рисунках 21 и 22 приведена возможная схема рабочего места сотрудника.

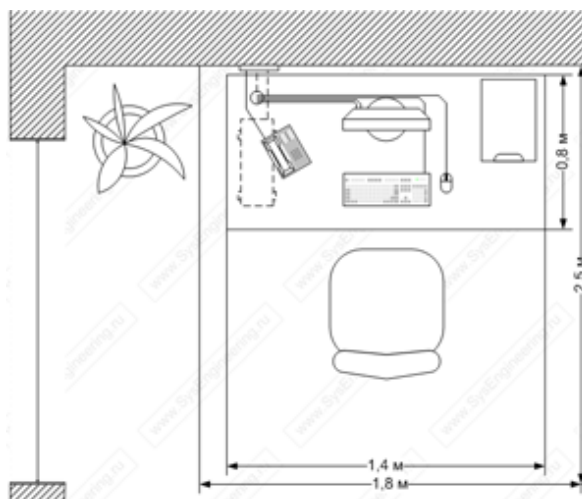


Рисунок 21. Схема рабочего места (вид сверху)

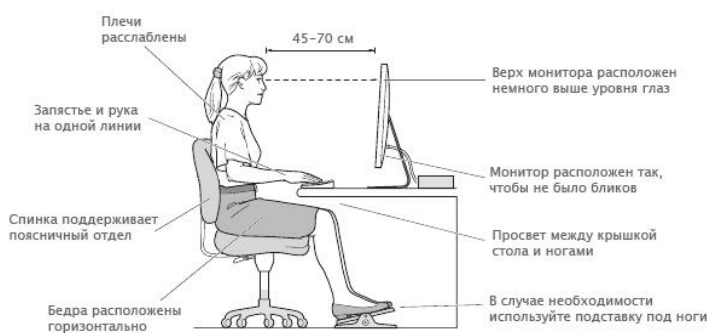


Рисунок 22. Схема рабочего места (вид сбоку)

Согласно 212 статье ТК РФ работодатель имеет ряд обязательств по обеспечению безопасных условий и охраны труда. В данный ряд входит обеспечение соответствия условий рабочих мест и режима работы требованиям норм охраны труда, проведение медицинских осмотров, организация проведения медицинских осмотров, установление режима труда и отдыха в соответствии с законодательством и т.д.

6.5 Заключение по разделу

В ходе выполнения работы над разделом «Социальная ответственность» были выявлены опасные и вредные факторы, воздействию которых может подвергнуться человек, разрабатывающий алгоритм построения карт глубины с

применением нейронных сетей. Был проведен анализ нормативной документации.

В целом, рабочее место удовлетворяет требованиям безопасности. Выполняемая работа не сопряжена с высоким риском травматизма.

Освещение на рабочем месте соответствует нормам – используется несколько энергосберегающих ламп.

Уровни шума находятся в допустимых пределах – источником шума при эксплуатации ПК могут являться системы охлаждения, а также жесткий диск, однако уровень создаваемого ими шума невысок.

Микроклиматические условия соблюдаются за счет использования систем отопления и кондиционирования.

Защита от повреждений электроники статическим электричеством не обеспечивается, однако так как корпус ПК закрыт, вероятность поражения элементов или работника минимальна (если не прикасаться мокрыми руками к корпусу).

Во время работы делаются перерывы для снижения нагрузки и предотвращения нервно-психических перегрузок.

Помещение оборудовано согласно требованиям электробезопасности.

В случае выхода из строя используемой электроники или ламп, отходы передаются в соответствующие компании.

Рабочее помещение оборудовано в соответствии с требованиями пожарной безопасности. Имеется порошковый огнетушитель, а также пожарная сигнализация.

Заключение

Основной целью работы была разработка программной компоненты для построения карт глубины по стереоизображениям.

Для достижения этой цели перед исполнителем была поставлена задача разработки алгоритма машинного обучения для построения карт глубины. Существующие методы поиска объектов на изображениях были изучены, и был реализован соответствующий алгоритм, позволяющий решить эту задачу.

Итоговая программа способна строить карты глубины, которые могут использоваться в других приложениях.

Естественно, в реализации остаются места, которые возможно оптимизировать для лучшей работы, в частности, улучшить качество работы алгоритма путем настройки гиперпараметров модели и проверить работу алгоритма на реальных данных, однако основная часть программы работает без накладок.

Список публикаций студента

1. Aleksandrova T. V., Gromakov E. I., Chugunov R. A., Syryamkin V. I. Virtual Analysis of Technical Equipment Operational States Variability // MATEC Web of Conferences. - 2016 - Vol. 79, Article Number 01006. - p. 1-6

Список использованных источников

1. Real-Time Navigation in 3D Environments Based on Depth Camera Data, Daniel Maier, Armin Hornung, Maren Bennewitz; [Электронный ресурс]

URL: <http://www2.informatik.unifreiburg.de/~hornunga/pub/maier12humanoids.pdf>
(Дата обращения: 30.03.2018).

2. A Review on Depth Estimation for Computer Vision Applications, Dineesh Mohan, Dr. A. Ranjith Ram, International Journal of Engineering and Innovative Technology (IJEIT) Volume 4, Issue 11, May 2015

3. Википедия. Глубокое обучение. [Электронный ресурс].

URL: https://ru.wikipedia.org/wiki/%D0%93%D0%BB%D1%83%D0%B1%D0%BE%D0%BA%D0%BE%D0%B5_%D0%BE%D0%B1%D1%83%D1%87%D0%B5%D0%BD%D0%B8%D0%B5

(Дата обращения: 30.03.2018).

4. Arxiv. [Электронный ресурс].

URL: <https://arxiv.org/pdf/1710.09829.pdf>

(Дата обращения: 30.03.2018).

5. ГОСТ 12.0.003-74 «Опасные и вредные производственные факторы. Классификация».

6. СанПиН 2.2.2/2.4.1340-03.

7. ГОСТ 12.0.003-74 «Опасные и вредные производственные факторы. Классификация».

8. СанПиН 2.2.2/2.4.1340-03.

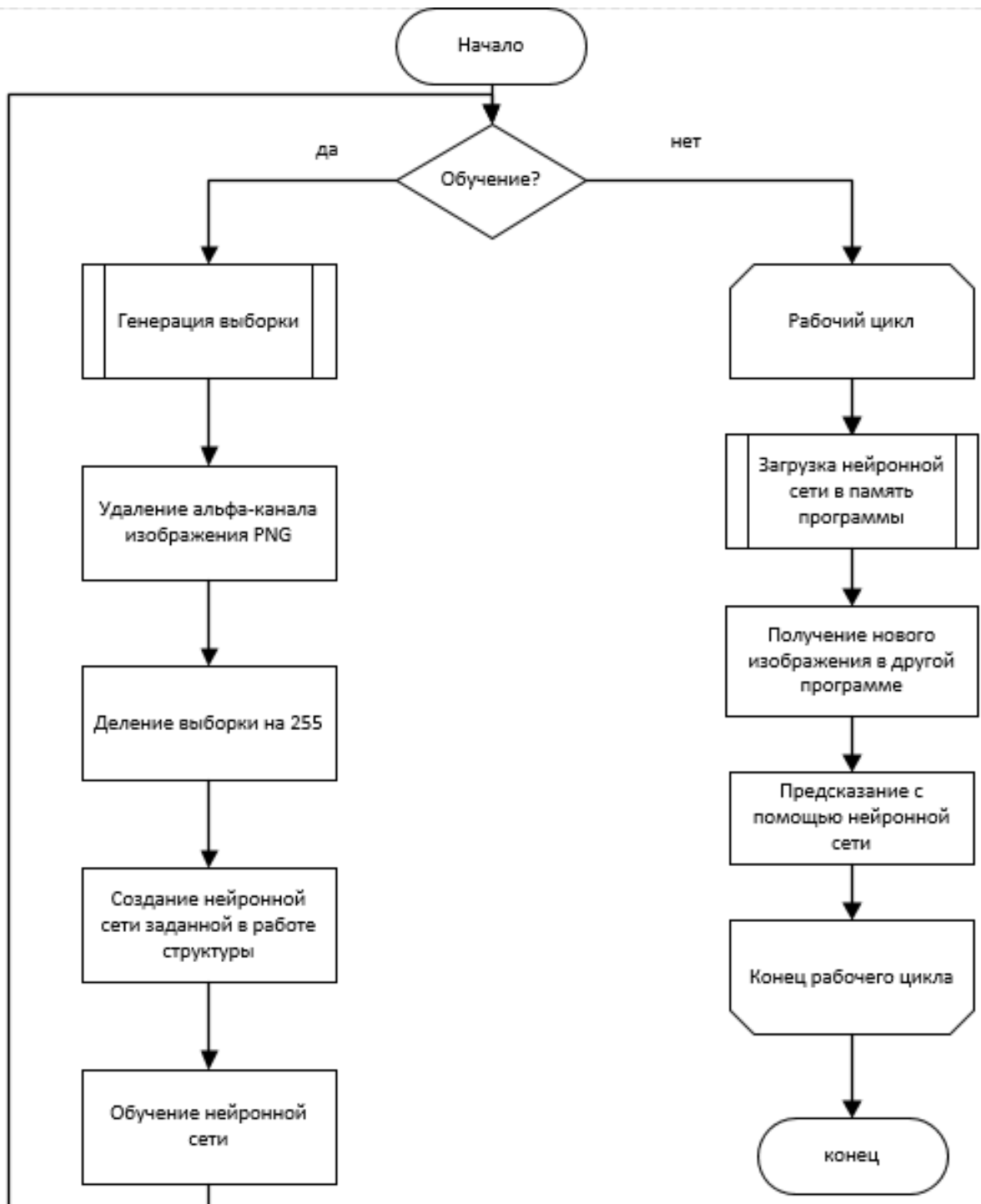
9. СанПиН 2.2.4.548-96

10. ТОИ Р-45-084-01

11. «Правила устройства электроустановок»
12. «Правила технической эксплуатации электроустановок потребителей»
13. «Межотраслевые правила охраны труда (правила безопасности) при эксплуатации электроустановок»
14. ГОСТ Р 53692-2009. «Ресурсосбережение. Обращение с отходами.»
15. Федеральный закон № 89 от 1998г. «Об отходах производства и потребления»
16. Постановление Правительства РФ № 340 от 2002 г.
17. ГОСТ 12.1.033-81 Система стандартов безопасности труда (ССБТ).
18. Пожарная безопасность. Термины и определения.
19. Трудовой Кодекс РФ

Приложения

Приложение А. Блок-схема алгоритма





Приложение Б. Исходный код

```
import bpy
from bpy import context
from math import sin, cos, radians
import random as rand
import time
#Output resolution (Stereoscopic images & depthmap)
bpy.context.scene.render.resolution_x = 320
bpy.context.scene.render.resolution_y = 240

# Total number of set of stereoscopic images and depth maps
total_scene_number = 10000

#####
#Start iteration to generate scenes
#####

def CreateCube(CubeName, MatName):
    bpy.ops.mesh.primitive_cube_add(location=((0.3-rand.random())*magn,
                                             (0.3-rand.random())*magn, (0.3-
rand.random())*magn))
    bpy.ops.transform.rotate(value=rand.random()*3.14, axis=(rand.random(),
                                                             rand.random(),
                                                             rand.random()),
                             constraint_axis=(False, False, False),
                             constraint_orientation='GLOBAL', mirror=False,
                             proportional='DISABLED',
                             proportional_edit_falloff='SMOOTH', proportional_size=1)
    bpy.ops.material.new()
    mat1 = bpy.data.materials[MatName]
    mat1.diffuse_color = (rand.random(), rand.random(), rand.random())

    bpy.data.objects[CubeName].data.materials.append(mat1)

def CreateCone(ConeName, MatName):
    bpy.ops.mesh.primitive_cone_add(location=((0.5-rand.random())*magn,
                                             (0.4-rand.random())*magn, (0.6-
rand.random())*magn))
    bpy.ops.transform.rotate(value=rand.random()*3.14,
                             axis=(rand.random(), rand.random(), rand.random()),
                             constraint_axis=(False, False, False),
                             constraint_orientation='GLOBAL',
                             mirror=False, proportional='DISABLED',
                             proportional_edit_falloff='SMOOTH',
                             proportional_size=1)
    bpy.ops.material.new()
    mat1 = bpy.data.materials[MatName]
    mat1.diffuse_color = (rand.random(), rand.random(), rand.random())

    bpy.data.objects[ConeName].data.materials.append(mat1)

def CreateTorus(TorusName, MatName):
    bpy.ops.mesh.primitive_torus_add(location=((0.35-rand.random())*magn,
                                             (0.55-rand.random())*magn, (0.45-
rand.random())*magn))
```

```

    bpy.ops.transform.rotate(value=rand.random()*3.14, axis=(rand.random(),
rand.random(), rand.random()),
                            constraint_axis=(False, False, False),
constraint_orientation='GLOBAL', mirror=False,
                            proportional='DISABLED',
proportional_edit_falloff='SMOOTH', proportional_size=1)
    bpy.ops.material.new()
    mat1 = bpy.data.materials[MatName]
    mat1.diffuse_color = (rand.random(), rand.random(), rand.random())

    bpy.data.objects[TorusName].data.materials.append(mat1)

def CreateSphere(SphereName, MatName):
    bpy.ops.mesh.primitive_uv_sphere_add(location=((0.5-rand.random())*magn,
                                                    (0.5-rand.random())*magn, (0.5-
rand.random())*magn))
    bpy.ops.transform.rotate(value=rand.random()*3.14, axis=(rand.random(),
rand.random(),
rand.random()),
                            constraint_axis=(False, False, False),
constraint_orientation='GLOBAL',
                            mirror=False, proportional='DISABLED',
proportional_edit_falloff='SMOOTH',
                            proportional_size=1)

    bpy.ops.material.new()
    mat1 = bpy.data.materials[MatName]
    mat1.diffuse_color = (rand.random(), rand.random(), rand.random())

    bpy.data.objects[SphereName].data.materials.append(mat1)

ii = 0

while ii < total_scene_number:

    ii += 1
    #Clear data from previous scenes
    for material in bpy.data.materials:
        material.user_clear();
        bpy.data.materials.remove(material);

    for texture in bpy.data.textures:
        texture.user_clear();
        bpy.data.textures.remove(texture);

    bpy.context.scene.use_nodes = True
    tree = bpy.context.scene.node_tree
    links = tree.links

    # clear default nodes
    for n in tree.nodes:
        tree.nodes.remove(n)

    #setup lighting:
    light = bpy.data.objects['Lamp']
    #light.data.shadow_method = 'RAY_SHADOW'
    light.data.energy = 5.0
    light.select = False

```

```

#setup camera:
camera = bpy.data.objects['Camera']
camera.select = True
camera.rotation_mode = 'XYZ'
angle1 = 1.3 + (0.5-rand.random())*1.5
angle2 = (0.5-rand.random())*1.5
angle3 = 0.75 + (0.5-rand.random())*1.5
camera.rotation_euler = (angle1, angle2, angle3)
Cam_x = 10 + (0.5-rand.random())*3
Cam_y = -3 + (0.5-rand.random())*3
Cam_z = 3 + (0.5-rand.random())*3
camera.location = (Cam_x,Cam_y,Cam_z)
camera.data.stereo.convergence_distance = 10000
camera.data.lens = 15 #(focal length)
camera.data.stereo.interocular_distance = 0.3
dist = ((camera.location[0]-(-3.22))**(2)+(camera.location[1]-
(8.0))**(2)+(camera.location[2]-(-5.425))**(2))**(1/2)
camera.select = False

#Remove objects from previous scenes
for item in bpy.data.objects:
    if item.type == "MESH":
        bpy.data.objects[item.name].select = True
        bpy.ops.object.delete()

for item in bpy.data.meshes:
    bpy.data.meshes.remove(item)

#####
#Create new scene:
#####

scene = bpy.context.scene
scene.render.use_multiview = True
scene.render.views_format = 'STEREO_3D'
rl = tree.nodes.new(type="CompositorNodeRLayers")
composite = tree.nodes.new(type = "CompositorNodeComposite")
composite.location = 200,0

scene = bpy.context.scene

#setup the depthmap calculation using blender's mist function:
scene.render.layers['RenderLayer'].use_pass_mist = True
scene.world.mist_settings.falloff = 'LINEAR'
#minimum depth:
scene.world.mist_settings.intensity = 0.0
scene.world.mist_settings.depth = dist
print(dist)

#magnitude of the random variation of object placements:
magn = 10;

#create objects with random location, orientation and color

CreateCube('Cube', 'Material')

CreateCube('Cube.001', 'Material.001')

CreateCone('Cone', 'Material.002')

```

```

CreateTorus('Torus', 'Material.003')

CreateSphere('Sphere', 'Material.004')

#create walls

bpy.ops.mesh.primitive_plane_add(radius=1, enter_editmode=False, location=(7,
8, -1), rotation=(0, 0, 0),
                                layers=(True, False, False, False, False,
False, False, False, False, False,
                                False, False, False, False, False,
False, False, False, False, False))
bpy.ops.transform.rotate(value=1.5708, axis=(0, 1, 0),
constraint_axis=(False, True, False),
constraint_orientation='GLOBAL', mirror=False,
proportional='DISABLED',
proportional_edit_falloff='SMOOTH',
proportional_size=1)
bpy.ops.transform.rotate(value=1.5708, axis=(0, 0, 1),
constraint_axis=(False, False, True),
constraint_orientation='GLOBAL', mirror=False,
proportional='DISABLED',
proportional_edit_falloff='SMOOTH',
proportional_size=1)

bpy.ops.transform.resize(value=(30, 30, 30), constraint_axis=(False, False,
False),
constraint_orientation='GLOBAL', mirror=False,
proportional='DISABLED',
proportional_edit_falloff='SMOOTH', proportional_size=1)

bpy.ops.mesh.primitive_plane_add(radius=1, enter_editmode=False, location=(-
3, -2, -1),
rotation=(0, 0, 0),
layers=(True, False, False, False, False,
False, False, False, False, False,
False, False, False, False, False,
False, False, False, False, False))
bpy.ops.transform.rotate(value=1.5708, axis=(0, 1, 0),
constraint_axis=(False, True, False),
constraint_orientation='GLOBAL', mirror=False,
proportional='DISABLED',
proportional_edit_falloff='SMOOTH',
proportional_size=1)
bpy.ops.transform.rotate(value=3.14, axis=(0, 0, 1), constraint_axis=(False,
False, True),
constraint_orientation='GLOBAL', mirror=False,
proportional='DISABLED',
proportional_edit_falloff='SMOOTH',
proportional_size=1)

bpy.ops.transform.resize(value=(30, 30, 30), constraint_axis=(False, False,
False),
constraint_orientation='GLOBAL', mirror=False,
proportional='DISABLED',
proportional_edit_falloff='SMOOTH',
proportional_size=1)

```

```

    bpy.ops.mesh.primitive_plane_add(radius=1, enter_editmode=False, location=(-
3, -2, -1),
                                rotation=(0, 0, 0),
                                layers=(True, False, False, False, False,
False, False, False, False, False,
                                False, False, False, False, False,
False, False, False, False, False))
    bpy.ops.transform.rotate(value=1.5708, axis=(0, 1, 0),
constraint_axis=(False, True, False),
                                constraint_orientation='GLOBAL', mirror=False,
proportional='DISABLED',
                                proportional_edit_falloff='SMOOTH',
proportional_size=1)
    bpy.ops.transform.rotate(value=3.14, axis=(0, 0, 1), constraint_axis=(False,
False, True),
                                constraint_orientation='GLOBAL', mirror=False,
proportional='DISABLED',
                                proportional_edit_falloff='SMOOTH',
proportional_size=1)

    bpy.ops.transform.resize(value=(30, 30, 30), constraint_axis=(False, False,
False),
                                constraint_orientation='GLOBAL', mirror=False,
proportional='DISABLED',
                                proportional_edit_falloff='SMOOTH',
proportional_size=1)

    bpy.ops.mesh.primitive_plane_add(location=(3.5, 0, -5), rotation=(0, 0, 0))
    bpy.ops.transform.resize(value=(30, 30, 30), constraint_axis=(False, False,
False),
                                constraint_orientation='GLOBAL', mirror=False,
proportional='DISABLED',
                                proportional_edit_falloff='SMOOTH',
proportional_size=1)

    bpy.ops.mesh.primitive_plane_add(location=(3.5, 0, 9), rotation=(0, 0, 0))
    bpy.ops.transform.resize(value=(30, 30, 30), constraint_axis=(False, False,
False),
                                constraint_orientation='GLOBAL', mirror=False,
proportional='DISABLED',
                                proportional_edit_falloff='SMOOTH',
proportional_size=1)

    bpy.ops.material.new()
    mat3 = bpy.data.materials['Material.005']
    bpy.ops.texture.new()
    texture1 = bpy.data.textures["Texture"].type = 'VORONOI'
    bpy.data.textures["Texture"].noise_scale = 0.02

    bpy.data.objects['Plane'].data.materials.append(mat3)
    bpy.data.objects['Plane.001'].data.materials.append(mat3)
    bpy.data.objects['Plane.002'].data.materials.append(mat3)
    bpy.data.objects['Plane.003'].data.materials.append(mat3)
    #####
    #Render the scene
    #####

    #ouput the depthmap:
    #####

```

```

#output the depthmap:
links.new(rl.outputs['Mist'], composite.inputs['Image'])

scene.render.use_multiview = False

scene.render.filepath = 'Depth_map/DepthMap_'+str(ii)+'.png'
bpy.ops.render.render( write_still=True )
#time.sleep(0.2)
#output the stereoscopic images:
links.new(rl.outputs['Image'], composite.inputs['Image'])

scene.render.use_multiview = True

scene.render.filepath = 'StereoImages/Stereoscopic_'+str(ii)+'.png'
bpy.ops.render.render( write_still=True )
#time.sleep(0.2)

# coding: utf-8
import os
import random
import warnings
import numpy as np
import matplotlib.pyplot as plt
from keras.models import Model, load_model
from keras.layers import Input
from keras.layers.core import Lambda
from keras.layers.convolutional import Conv2D, Conv2DTranspose
from keras.layers.pooling import MaxPooling2D
from keras.layers.merge import concatenate
from keras.layers import BatchNormalization
from keras.callbacks import EarlyStopping, ModelCheckpoint
from PIL import Image
from sklearn.model_selection import train_test_split
from sklearn.metrics import mean_squared_error
from keras import backend as K
from keras.models import load_model
import cv2
import tensorflow as tf
import time

# Set some parameters
IMG_WIDTH = 640
IMG_HEIGHT = 240
IMG_CHANNELS = 3
smooth = 1.
warnings.filterwarnings('ignore', category=UserWarning, module='skimage')
# seed = 42
# random.seed = seed
# np.random.seed = seed
input_path = os.path.dirname(os.path.abspath(__file__))

# модель
def keras_model():
    neurons = 16
    # Build U-Net model
    inputs = Input((IMG_HEIGHT, IMG_WIDTH, IMG_CHANNELS))
    s = Lambda(lambda x: x / 255.) (inputs)
    c0 = Conv2D(neurons, (3, 3), activation='elu', padding='same')(s)

```



```

c0 = BatchNormalization() (c0)
c0 = Conv2D(neurons, (3, 3), activation='elu', padding='same') (c0)
c0 = BatchNormalization() (c0)
p0 = MaxPooling2D((1, 2)) (c0)
p0 = BatchNormalization() (p0)

c1 = Conv2D(neurons, (3, 3), activation='elu', padding='same') (p0)
c1 = BatchNormalization() (c1)
c1 = Conv2D(neurons, (3, 3), activation='elu', padding='same') (c1)
c1 = BatchNormalization() (c1)
p1 = MaxPooling2D((2, 2)) (c1)
p1 = BatchNormalization() (p1)

c2 = Conv2D(neurons*2, (3, 3), activation='elu', padding='same') (p1)
c2 = BatchNormalization() (c2)
c2 = Conv2D(neurons*2, (3, 3), activation='elu', padding='same') (c2)
c2 = BatchNormalization() (c2)
p2 = MaxPooling2D((2, 2)) (c2)
p2 = BatchNormalization() (p2)

c3 = Conv2D(neurons*4, (3, 3), activation='elu', padding='same') (p2)
c3 = BatchNormalization() (c3)
c3 = Conv2D(neurons*4, (3, 3), activation='elu', padding='same') (c3)
c3 = BatchNormalization() (c3)
p3 = MaxPooling2D((2, 2)) (c3)
p3 = BatchNormalization() (p3)

c4 = Conv2D(neurons*8, (3, 3), activation='elu', padding='same') (p3)
c4 = BatchNormalization() (c4)
c4 = Conv2D(neurons*8, (3, 3), activation='elu', padding='same') (c4)
c4 = BatchNormalization() (c4)
p4 = MaxPooling2D(pool_size=(2, 2)) (c4)
p4 = BatchNormalization() (p4)

c5 = Conv2D(neurons*16, (3, 3), activation='elu', padding='same') (p4)
c5 = BatchNormalization() (c5)
c5 = Conv2D(neurons*16, (3, 3), activation='elu', padding='same') (c5)
c5 = BatchNormalization() (c5)

u6 = Conv2DTranspose(neurons*8, (2, 2), strides=(2, 2), padding='same') (c5)
u6 = BatchNormalization() (u6)
u6 = concatenate([u6, c4])
u6 = BatchNormalization() (u6)
c6 = Conv2D(neurons*8, (3, 3), activation='elu', padding='same') (u6)
c6 = BatchNormalization() (c6)
c6 = Conv2D(neurons*8, (3, 3), activation='elu', padding='same') (c6)
c6 = BatchNormalization() (c6)

u7 = Conv2DTranspose(neurons*4, (2, 2), strides=(2, 2), padding='same') (c6)
u7 = BatchNormalization() (u7)
u7 = concatenate([u7, c3])
u7 = BatchNormalization() (u7)
c7 = Conv2D(neurons*4, (3, 3), activation='elu', padding='same') (u7)
c7 = BatchNormalization() (c7)
c7 = Conv2D(neurons*4, (3, 3), activation='elu', padding='same') (c7)
c7 = BatchNormalization() (c7)

u8 = Conv2DTranspose(neurons*2, (2, 2), strides=(2, 2), padding='same') (c7)
u8 = BatchNormalization() (u8)

```

```

u8 = concatenate([u8, c2])
u8 = BatchNormalization() (u8)
c8 = Conv2D(neurons*2, (3, 3), activation='elu', padding='same') (u8)
c8 = BatchNormalization() (c8)
c8 = Conv2D(neurons*2, (3, 3), activation='elu', padding='same') (c8)
c8 = BatchNormalization() (c8)

u9 = Conv2DTranspose(neurons, (2, 2), strides=(2, 2), padding='same') (c8)
u9 = BatchNormalization() (u9)
u9 = concatenate([u9, c1], axis=3)
u9 = BatchNormalization() (u9)
c9 = Conv2D(neurons, (3, 3), activation='elu', padding='same') (u9)
c9 = BatchNormalization() (c9)
c9 = Conv2D(neurons, (3, 3), activation='elu', padding='same') (c9)
c9 = BatchNormalization() (c9)

outputs = Conv2D(1, (1, 1), activation='sigmoid') (c9)
model = Model(inputs=[inputs], outputs=[outputs])
model.compile(optimizer='adam', loss='mean_squared_error')
model.summary()
return model

# Преобразование изображения с альфа-каналом в RGB
def pure_pil_alpha_to_color(image, color=(255, 255, 255)):
    image.load() # needed for split()
    background = Image.new('RGB', image.size, color)
    background.paste(image, mask=image.split()[3]) # 3 is the alpha channel
    return background

# RGB в grayscale
def rgb2gray(rgb):
    return np.dot(rgb[...,:3], [0.299, 0.587, 0.114])

# Загрузка датасета
def load_images(limit=7000):
    depth_maps = os.listdir(input_path + '/Depth_map/')
    depth_maps = [input_path + '/Depth_map/' + file for file in depth_maps]
    x = []
    y = []
    print('Reading X...')
    for i, file in enumerate(depth_maps):
        if i >= limit:
            break
        with open(file, 'rb') as f:
            depthM = Image.open(f)
            depthM = pure_pil_alpha_to_color(depthM)
            depthM = np.array(depthM, dtype=np.uint8)
            depthM = rgb2gray(depthM)
            depthM = np.expand_dims(depthM, axis=-1)
        y.append(depthM)
    y = np.array(y).astype(np.float32)
    stereo = os.listdir(input_path + '/StereoImages/')
    stereo = [input_path + '/StereoImages/' + file for file in stereo]
    print('Reading y...')
    for i, file in enumerate(stereo):
        if i >= limit:
            break
        with open(file, 'rb') as f:
            stereo_image = Image.open(f)
            stereo_image = pure_pil_alpha_to_color(stereo_image)

```

```

        stereo_image = np.array(stereo_image, dtype=np.uint8)
        x.append(stereo_image)
    x = np.array(x)
    y = y/255.
    return x, y

depth_maps = os.listdir(input_path + '/Depth_map/')
depth_maps = [input_path + '/Depth_map/' + file for file in depth_maps]
x, y = load_images()
print(np.max(x))

print(x.shape)
print(y.shape)

train = False
model = keras_model()
x_train, x_test, y_train, y_test = train_test_split(x, y, test_size=0.1)
earlystopper = EarlyStopping(patience=5, verbose=1)
checkpointer = ModelCheckpoint('model-keras.h5', verbose=1, save_best_only=True)

if train:
    results = model.fit(x_train, y_train, validation_split=0.1, batch_size=16,
epochs=1000,
                        callbacks=[earlystopper, checkpointer])
model = load_model('model-keras.h5')

test_predict = model.predict(x_test)

print('MSE neural: {}'.format(mean_squared_error(np.reshape(y_test,
(y_test.shape[0], -1)), np.reshape(test_predict, (test_predict.shape[0], -1)))))

start = time.clock()
model.predict(np.expand_dims(x_test[0], axis=0))
stop = time.clock()
print('Neural network GPU: {} seconds'.format(stop-start))
with tf.device('/cpu:0'):
    start = time.clock()
    model.predict(np.expand_dims(x_test[0], axis=0))
    stop = time.clock()
    print('Neural network CPU: {} seconds'.format(stop - start))

left_im, right_im = cv2.cvtColor(((x_test[0, :, :320] * 255.).astype(np.uint8)),
cv2.COLOR_BGR2GRAY), \
                    cv2.cvtColor(((x_test[0, :, 320:] * 255.).astype(np.uint8)),
cv2.COLOR_BGR2GRAY)

start = time.clock()
stereo = cv2.StereoBM_create(numDisparities =16, blockSize=15)
start_1 = time.clock()
disparity = stereo.compute(left_im, right_im)
stop = time.clock()
print('OpenCV: {} seconds'.format(stop-start))
print('OpenCV v2: {} seconds'.format(stop-start_1))
test_predict_opencv = np.zeros(test_predict.shape)

for i, image in enumerate(x_test):
    left_im, right_im = cv2.cvtColor(((x_test[i, :, :320] *
255.).astype(np.uint8)), cv2.COLOR_BGR2GRAY), \
                            cv2.cvtColor(((x_test[i, :, 320:] *
255.).astype(np.uint8)), cv2.COLOR_BGR2GRAY)

```

```

    disparity = stereo.compute(left_im, right_im)
    test_predict_opencv[i] = np.expand_dims(disparity, axis=-1)

print('MSE OpenCV: {}'.format(mean_squared_error(np.reshape(y_test,
(y_test.shape[0], -1)),
np.reshape(test_predict_opencv/255., (test_predict_opencv.shape[0], -1))))

# сохранение пикчей
for i in range(100):
    plt.imshow(np.squeeze(test_predict[i]))
    plt.savefig('_prediction/pred_{}.jpg'.format(str(i)))
    plt.imshow(np.squeeze(y_test[i]))
    plt.savefig('_true/true_{}.jpg'.format(str(i)))
    plt.imshow(np.squeeze(np.clip(y_test[i] - test_predict[i], 0., 1.)))
    plt.savefig('_diff/diff_true_minus_pred{}.jpg'.format(str(i)))
    plt.imshow(np.squeeze(np.clip(test_predict[i] - y_test[i], 0., 1.)))
    plt.savefig('_diff/diff_pred_minus_true{}.jpg'.format(str(i)))
    left_im, right_im = cv2.cvtColor((x_test[i, :, :320] *
255.).astype(np.uint8)), cv2.COLOR_BGR2GRAY), \
        cv2.cvtColor((x_test[i, :, 320:] *
255.).astype(np.uint8)), cv2.COLOR_BGR2GRAY)
    disparity = stereo.compute(left_im, right_im)
    plt.imshow(disparity)
    plt.savefig('_opencv/cv2_{}.jpg'.format(str(i)))

```