Министерство науки и высшего образования Российской Федерации

Федеральное государственное автономное образовательное учреждение

высшего образования

УРАЛЬСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ ИМЕНИ ПЕРВОГО ПРЕЗИДЕНТА РОССИИ Б.Н. ЕЛЬЦИНА

(УрФУ)

Физико-технологический институт

Кафедра экспериментальной физики

ДОПУСТИТЬ К ЗАЩИТЕ В ГЭК

Зав. кафедрой ЭФ

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ В.Ю. Иванов

( подпись)

«\_\_\_\_\_\_»\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_2019 г.

**СИСТЕМА МАШИННОГО ЗРЕНИЯ ДЛЯ КОНТРОЛЯ МЕССБАУЭРОВСКОГО СПЕКТРОМЕТРИЧЕСКОГО КОМПЛЕКСА**

ДИПЛОМНАЯ РАБОТА

Пояснительная записка

14.05.04 62.01.11 224 ПЗ

|  |  |  |
| --- | --- | --- |
| Руководитель  доц., к.ф.-м.н. | \_\_\_\_\_\_\_\_\_\_\_\_\_ | М.В. Ушаков |
|  |  |  |
| Консультант  доц., к.ф.-м.н. | \_\_\_\_\_\_\_\_\_\_\_\_\_ | А.А. Екидин |
|  |  |  |
| Консультант  доц., к.т.н. | \_\_\_\_\_\_\_\_\_\_\_\_\_ | Ю.В. Волкова |
|  |  |  |
| Нормоконтролер  вед. инж. | \_\_\_\_\_\_\_\_\_\_\_\_\_ | Ю.Г. Лазарев |
|  |  |  |
| Студент гр. Фт-540301 | \_\_\_\_\_\_\_\_\_\_\_\_\_ | Г.Д. Серяков |

Екатеринбург

2019

# РЕФЕРАТ

Пояснительная записка 70 с., 6 разделов, 17 рис., 1 табл., 32 ист.

НЕЙРОСЕТИ, АВТОМАТИЗАЦИЯ, ФИЗИЧЕСКИЕ УСТАНОВКИ, МАШИННОЕ ЗРЕНИЕ, МЕССБАУЭРОВСКАЯ СПЕКТРОСКОПИЯ, TENSORFLOW, KERAS, PYTHON

Цель работы – разработка системы машинного зрения для контроля мессбауэровского спектрометрического комплекса.

В ходе работы были изучены принципы построения нейронных сетей, возможности их применения и существующие библиотеки для их создания. , разработана архитектура автоматического контроля комплекса, созданы нейронные сети для контроля работы источников питания детекторов и определения показаний манометра. Использовался язык программирования Python, библиотеки для глубокого обучения Keras и TensorFlow, а также графический редактор Photoshop для подготовки базы данных для обучения сетей.

В результате выполнения работы была разработана архитектура системы контроля спектрометрического комплекса и выбраны технические решения для ее реализации. Были предложены архитектуры двух нейронных сетей для контроля состояния источника питания детекторов МКП 011 и криостата мессбауэровского спектрометрического комплекса.

# СОДЕРЖАНИЕ

[ПЕРЕЧЕНЬ СОКРАЩЕНИЙ И ОБОЗНАЧЕНИЙ 5](#_Toc11683702)

[ВВЕДЕНИЕ 6](#_Toc11683703)

[1 Нейронные сети 9](#_Toc11683704)

[1.1 Принципы построения и обучения НС 9](#_Toc11683705)

[1.2 Известные типы нейронных сетей 14](#_Toc11683706)

[1.3 Промышленное применение нейронных сетей 16](#_Toc11683707)

[1.4 Библиотеки и фреймворки для нейронных сетей на языке Python 19](#_Toc11683708)

[1.5 Формулирование технического задания на разработку системы 22](#_Toc11683709)

[2 Нейронная сеть для контроля работы источника питания детекторов 23](#_Toc11683710)

[3 Нейронная сеть для определения показаний давления 29](#_Toc11683711)

[4 Нейронная сеть для определения местоположений областей интереса на изображении 35](#_Toc11683712)

[5 Безопасность жизнедеятельности 36](#_Toc11683713)

[5.1 Описание объекта дипломирования 36](#_Toc11683714)

[5.1.1 Описание рабочего места 37](#_Toc11683715)

[5.1.2 Интенсивность труда 38](#_Toc11683716)

[5.1.3 Микроклимат 38](#_Toc11683717)

[5.1.4 Освещенность 40](#_Toc11683718)

[5.1.5 Эргономичность 40](#_Toc11683719)

[5.2 Опасные производственные факторы 41](#_Toc11683720)

[5.2.1 Электробезопасность 41](#_Toc11683721)

[5.2.2 Вредные факторы и мероприятия по их устранению 42](#_Toc11683722)

[5.2.3 Пожарная охрана 43](#_Toc11683723)

[5.2.4 Возникновение ЧС 44](#_Toc11683724)

[5.2.5 Молниезащита 46](#_Toc11683725)

[5.3 Выводы 46](#_Toc11683726)

[6 Природопользование и охрана окружающей среды 47](#_Toc11683727)

[6.1 Введение 47](#_Toc11683728)

[6.2 Влияние объекта дипломной работы на окружающую среду 47](#_Toc11683729)

[6.3 Утилизация электронной техники 48](#_Toc11683730)

[6.4 Способы обеспечения экологической безопасности 49](#_Toc11683731)

[6.5 Потребление энергоресурсов 50](#_Toc11683732)

[6.6 Утилизация электронной техники и отходов инфраструктуры 50](#_Toc11683733)

[ЗАКЛЮЧЕНИЕ 53](#_Toc11683734)

[СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ 54](#_Toc11683735)

[ПРИЛОЖЕНИЕ А Различные варианты структур НС 59](#_Toc11683736)

[ПРИЛОЖЕНИЕ Б Структура нейронной сети для источника питания 60](#_Toc11683737)

[ПРИЛОЖЕНИЕ В Структура нейронной сети для манометра 61](#_Toc11683738)

[ПРИЛОЖЕНИЕ Г Листинг программы для конвертации файлов из формата .xml в .csv 62](#_Toc11683739)

[ПРИЛОЖЕНИЕ Д Листинг программы для конвертации файлов из формата .csv в .record 64](#_Toc11683740)

[ПРИЛОЖЕНИЕ Е Листинг программы для обучения НС 67](#_Toc11683741)

# ПЕРЕЧЕНЬ СОКРАЩЕНИЙ И ОБОЗНАЧЕНИЙ

АСУ – автоматическая система управления

ИПТ – источник питания детекторов

КЗ – короткое замыкание

НС – нейронная сеть

ПО – программное обеспеченье

# ВВЕДЕНИЕ

Контроль параметров работы физических и промышленных установок является наиважнейшим критерием получения корректных результатов работы на данных установках. Контроль особенно важен, когда процессы занимают длительный интервал времени и / или используется дорогостоящее сырье. Одной из таких установок является рассматриваемый в настоящей работе мессбауэровский спектрометрический комплекс, построенный на базе спектрометра СМ-2201 [1-3]. Он используется для получения физической информации об особенностях структуры и магнитных свойств различных объектов от наночастиц и метеоритов до биомолекул. При этом время измерения зависит от содержания ядер 57Fe в образцах и для обеспечения необходимой величины соотношения сигнал/шум (минимальное значение - 40) оно может составлять до 2-3 недель.

Этот спектрометрический комплекс обладает большим числом параметров, которые необходимо контролировать, в противном случае может произойти поломка с долговременным последующим запуском спектрометрического комплекса (время запуска может составлять до 6-8 месяцев) Одними из наиболее важных объектов контроля этого комплекса являются:

* высоковольтный источник питания детектора ионизирующего излучения;
* криостат для контроля температуры образца.

При отсутствии высокого напряжения питания на детекторе прекратится набор данных, что в случаи длительных экспериментов приведет к потере времени, в течение которого спектрометр работал с выключенным высоковольтным источником питания детектора.

При сбое в работе криостата возникает возможность аварии, связанная с разрушением цепи поступления паров азота в камеру с исследуемым образцом, что может привести, в лучшем случае, к отогреву и структурным изменениям исследуемого объекта, а в худшем случае, к попаданию жидкости в электронные модули спектрометра, что в свою очередь приведет к поломке и остановке всего комплекса.

Исходя из всего вышеперечисленного, возникает острая необходимость контролировать работу установки в режиме непрерывных измерений 24/7, эту задачу можно решать несколькими способами:

* использовать одного или нескольких операторов, которые будут вести визуальный контроль работы установки;
* подключить набор дополнительных датчиков с возможностью информирования о сбое в работе установке;
* использовать возможности программного обеспечения для получения параметров работы установки;
* использовать систему, построенную на основе машинного зрения.

Среди этих вариантов первый вариант не может быть использован так, как он является малоэффективным в силу большого вклада человеческого фактора. Второй вариант является нереализуемым, так как мессбауэровский спектрометрический комплекс является полностью законченной установкой и подключение дополнительных датчиков не предоставляется возможным. Третий вариант является оптимальным в отношении трудозатрат, но доступ к состоянию контролируемых объектов невозможно получить из ПО мессбауэровского спектрометрического комплекса, а расширение этого ПО также не представляется возможным (часть из ПО является проприетарным, а у другой части закрытые исходные тексты). Поэтому построение системы на основе машинного зрения остается единственным возможным вариантом для создания системы автоматизированного контроля мессбауэровского спектрометрического комплекса.

Машинное зрение [4, 5] — это научное направление в области искусственного интеллекта, в частности робототехники, и связанные с ним технологии получения изображений объектов реального мира, их обработки и использования, полученных данных для решения разного рода прикладных задач без участия (полного или частичного) человека. Машинное зрение сосредотачивается на применении, в основном промышленном, например, автономные роботы и системы визуальной проверки и измерений. Одним из возможных методов реализации машинного зрения является использование нейронной сети [6, 7], что является совокупностью математических моделей, воспроизводящих работу человеческого мозга для создания машин, обладающих искусственным интеллектом. Для того, чтобы нейронная сеть могла работать она должна быть обучена на определение ряда отличительных признаков. Существует несколько подходов к обучению нейронных сетей, одним из которых является обучение с учителем. Это означает наличие обучающего набора данных, который содержит примеры с истинными значениями: тегами, классами и показателями.

Контроль над источником питания детекторов можно осуществлять по светодиодной индикации (наличие выходного напряжения и перегрузки), расположенной на передней части прибора. Контроль над криостатом можно проводить посредством наблюдения за показанием стрелочного манометра (высокое и низкое значение давления паров азота в криостате будет свидетельствовать о наличии проблем в работе криостата).

В связи с этим цель данной работы – разработка системы машинного зрения для контроля мессбауэровского спектрометрического комплекса.

Для достижения этой цели были поставлены следующие задачи:

* разработать архитектуру системы контроля комплекса;
* построить структуры НС для контроля световой индикации работы источника питания детекторов и показаний манометра криостата;
* провести обучение и проверку НС на тестовом наборе данных.

# 1 Нейронные сети

## 1.1 Принципы построения и обучения НС

Искусственная нейронная сеть – набор связанных между собой нейронов, которые представляют собой элемент обработки информации. Как правило нейронная сеть состоит из большого числа нейронов, например, в работе [8] была предложена нейронная сеть для распознавания предметов одежды из набора данных Fashin MNIST, состоящая из 800 нейронов. Любая искусственная нейронная сеть состоит из трех компонентов:

* входной слой;
* скрытые (вычислительные) слои;
* выходной слой.

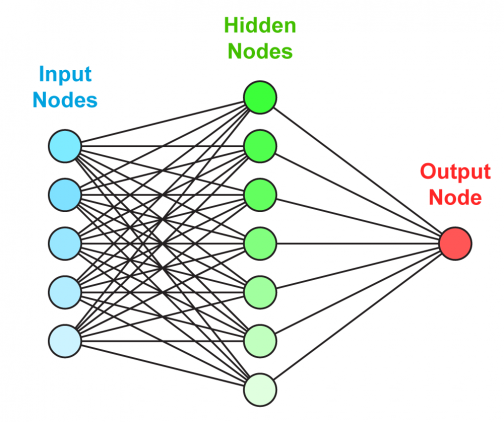


Рисунок 1.1 – Структура нейронной сети [7]

Нейроны представляют собой единицы обработки информации в нейронной сети. На рисунке ниже приведена модель нейрона, лежащего в основе искусственных нейронных сетей:

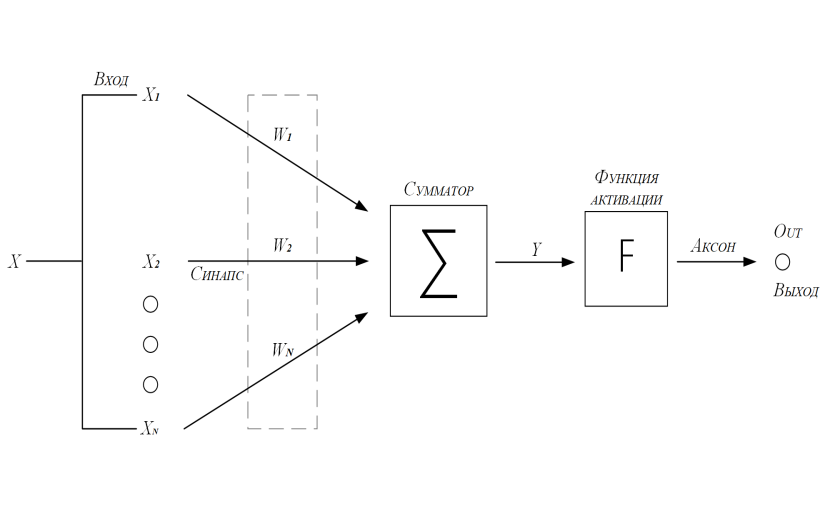


Рисунок 1.2 – Математическая модель нейрона

Модель нейрона имитирует в первом приближении свойства биологического нейрона. В этой модели можно выделить три основных элемента:

* синапсы, каждый из которых характеризуется своим силой или весом. Они осуществляют связь между нейронами, умножают входной сигнал x\_{i} на весовой коэффициент синапса w\_{i}, характеризующий силу синоптической связи;
* сумматор - аналог тела клетки нейрона. Выполняет сложение внешних входных сигналов или сигналов, поступающих по синоптическим связям от других нейронов, и определяет уровень возбуждения нейрона;
* функция активации определяет окончательный выходной уровень нейрона, с которым сигнал возбуждения (торможения) поступает на синапсы следующих нейронов.

Обучение НС происходит в два этапа:

* прямое распространение ошибки;
* обратное распространение ошибки.

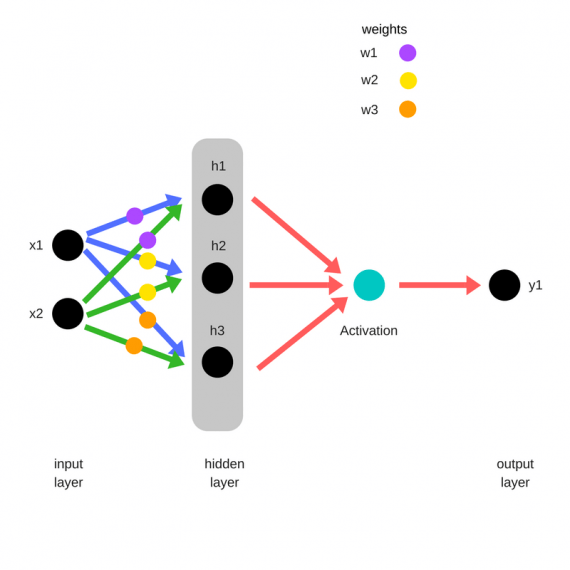


Рисунок 1.3 – Схема прямого распространения ошибки [7]

При прямом распространении начальные веса w1, w2, w3 задаются случайным образом и на них умножаются входные данные для формирования скрытого слоя:

(1.1)

(1.2)

(1.3)

где x1 и x2 – входные данные;

h1, h2 и h3 – данные скрытого слоя.

Выходные данные из скрытого слоя передается через нелинейную функцию ([функцию активации](https://neurohive.io/ru/osnovy-data-science/activation-functions/)), для получения выхода сети:

(1.4)

При обратном распространении ошибки вначале определяется суммарная ошибка (total\_error), которая вычисляется как разность между ожидаемым значением y (из обучающего набора) и полученным значением y\_, проходящих через функцию потерь (loss\_function).

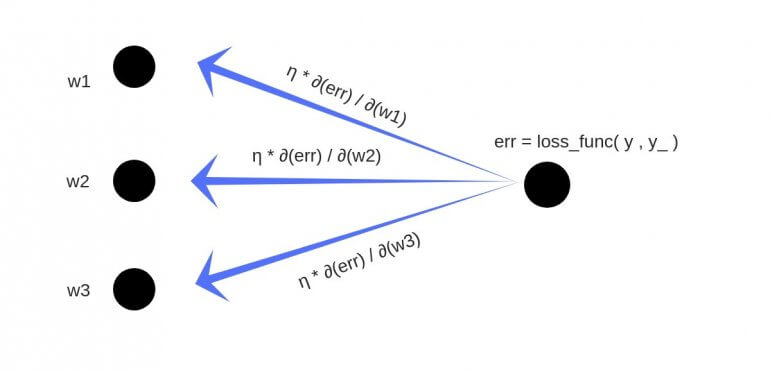


Рисунок 1.4 – Схема обратного распространения ошибки [7]

Далее вычисляется частная производная ошибки по каждому весу. Эти частные дифференциалы отражают вклад каждого веса в общую ошибку. Затем эти дифференциалы умножаются на число, называемое скорость обучения или learning rate (η).

Полученный результат затем вычитается из соответствующих весов. В результате получатся следующие обновленные веса:

(1.5)

(1.6)

, (1.7)

где err – суммарная ошибка.

Нейронная сеть используется для автоматизации отбора признаков, но некоторые параметры настраиваются вручную.

Скорость обучения является очень важным гиперпараметром. Если скорость обучения слишком мала, то даже после обучения НС в течение длительного времени она будет далека от оптимальных результатов. С другой стороны, если скорость обучения слишком высока, то сеть может «проскакивать» значения коэффициентов и тем самым никогда не достигнет требуемых результатов.

Функция активации — это один из самых мощных инструментов, который влияет на силу, приписываемую нейронным сетям. Отчасти, она определяет, какие нейроны будут активированы, и какая информация будет передаваться последующим слоям. Без функций активации глубокие сети теряют значительную часть своей способности к обучению. Нелинейность этих функций отвечает за повышение степени свободы, что позволяет обобщать проблемы высокой размерности в более низких измерениях. Ниже приведены примеры функций активации:

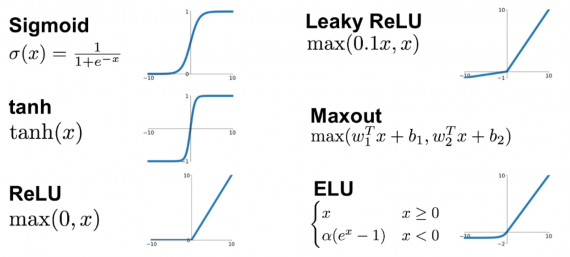


Рисунок 1.5 – Функции активации

Функция потерь находится в центре НС и она используется для расчета ошибки между реальными и полученными ответами. Основная задача процесса обучения – минимизация ошибки с помощью подбора оптимального значения функции потерь. Функция потерь оценивает насколько хорошо нейронная сеть работает в целом.

Некоторые известные функции потерь:

* квадратичная (среднеквадратичное отклонение);
* кросс-энтропия;
* экспоненциальная (AdaBoost);
* расстояние Кульбака — Лейблера или прирост информации.

Функция потерь в нейронной сети должна удовлетворять двум условиям:

* функция потерь должна быть записана как среднее;
* функция потерь не должна зависеть от каких-либо активационных значений нейронной сети, кроме значений, выдаваемых на выходе.

## 1.2 Известные типы нейронных сетей

Одним из наиболее оптимальных типов НС является сеть с глубоким обучением. Глубокое обучение (deep learning) – это класс алгоритмов [машинного обучения](https://neurohive.io/ru/osnovy-data-science/vvedenie-v-mashinnoe-obuchenie-kto-ego-primenjaet-i-kak-stat-razrabotchikom/), которые учатся более абстрактно понимать данные. Глубокие НС, в отличие от альтернативных подходов, могут работать со всем набором имеющихся данных. В процессе обучения НС сама определяет, какие признаки в данных для нее важны. Сеть может подобрать такое сложное сочетание признаков, до которого человек никогда бы не догадался. Поэтому с помощью глубоких нейронных сетей удается решать такие задачи, с которыми не справляются традиционные алгоритмы машинного обучения.

Глубокие НС используют каскад из множества обрабатывающих слоев (нелинейных) для извлечения и преобразования признаков. Принцип действия основывается на изучении признаков (представлении информации) в данных без обучения с учителем. Функции более высокого уровня (которые находятся в последних слоях) получаются из функций нижнего уровня (которые находятся в слоях начальных слоях). Глубокие НС изучают многоуровневые представления, которые соответствуют разным уровням

Из всего разнообразия глубоких НС, структуры которых лежат в открытом доступе, мы рассмотрим сети, разработанные для задач классификации изображения, такие как:

* VGG16 [9],
* VGG19 [9],
* ResNet50 [10].

VGG16 – сверточная нейронная сеть включающая 144 миллиона параметров, содержит 13 сверточных слоев с очень маленькими восприимчивыми полями (3 x 3), пять слоев подвыборки max-pooling размера 2 x 2 для выполнения пространственного объединения, а затем три полно -связанных слоя с последним слоем в качестве слоя soft-max. Функцию активации ReLu применили ко всем скрытым слоям. Модель также использует слои регуляризации Dropout для предотвращения переобучения НС.

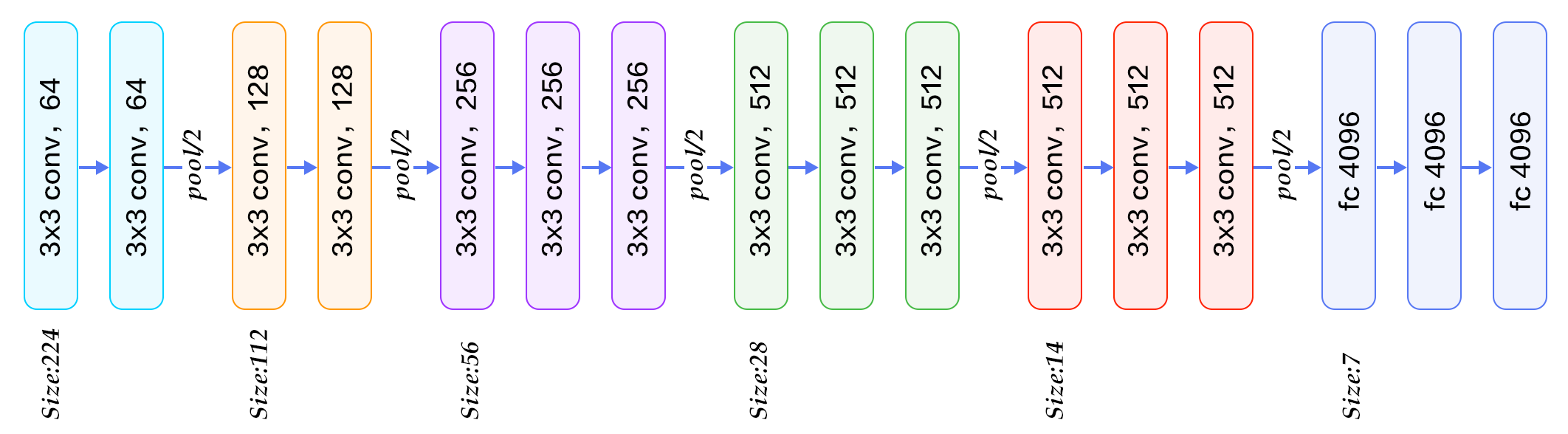


Рисунок 1.6 – Структура НС VGG16 [9]

VGG19 отличается от предыдущей НС наличием дополнительный 3 сверточных слоев.

В ResNet- 50 идет 48 сверточных слоев 2 полно-связных слоя.

## 1.3 Промышленное применение нейронных сетей

Преимущества НС делают их использование привлекательным для решения задач, таких как:

* прогнозирование;
* планирование;
* проектирование АСУ;
* управление качеством;
* управление манипуляторами и робототехникой;
* обеспечение безопасности производства: обнаружение неисправностей и предупреждение аварийных ситуаций;
* управление процессами: оптимизация режимов производственных процессов; мониторинг и визуализация диспетчерской информации[11].

Сегодня прогнозирование на основе НС наиболее полно реализовано в сфере экономики и финансов. В промышленном производстве НС могут быть полезными, например, при создании модели управления рисками предприятия, планировании производственного цикла. Моделирование и оптимизация производства характеризуется высокой сложностью, большим количеством переменных и констант, определенных не для всех возможных систем. Традиционные аналитические модели часто можно построить только при значительном упрощении, и они носят в основном оценочный характер. В то время как СН обучается на основе данных реального или численного эксперимента.

Классические методы построения АСУ технологическими процессами строятся на формализованных знаниях человека об объекте управления. Вариант построение АСУ на основе НС реализует свойственные человеку когнитивные приемы. В качестве примеров успешного применения НС в данной сфере можно назвать управление сложными процессами и объектами в условиях информационной неопределенности, процессами механообработки, робототехническими системами и др.

Большой опыт накоплен в области использования НС при управлении качеством в промышленности. Например, НС, примененная на предприятиях Intel, для идентификации брака при производстве микросхем способна забраковать неисправный чип с точностью 99,5 %. Путем подачи звуковых волн и приема отраженного сигнала, а затем обработкой НС, специалисты из National Institute of Standards and Technology (NIST) проверяют качество бетона при толщине материала до полуметра.

В области обнаружения неисправностей использование НС позволяет в режиме реального времени следить за состоянием оборудования, выявлять отклонения и предупреждать наступление аварийных ситуаций. Большие перспективы открывает использование НС в области мониторинга загрязнения окружающей среды, что также понижает риск техногенных аварий.

С помощью НС успешно решается важная задача в области телекоммуникаций – нахождение оптимального пути трафика между узлами. Учитываются две особенности: во-первых, решение должно быть адаптивным, т. е. учитывать текущее состояние сети связи и наличие сбойных участков, а во-вторых, оптимальное решение необходимо находить в реальном времени. Кроме управления маршрутизацией потоков, нейронные сети используются для получения эффективных решений при проектировании телекоммуникационных сетей.

В области обеспечения информационной безопасности НС также успешно решают задачи, недоступные традиционным подсистемам, ориентированным на заранее заданные классы угроз в условиях возрастающей сложности и динамики разнородного программного обеспечения. Так, хорошо исследованы возможности применения технологий на базе НС для идентификации и аутентификации, антивирусной защиты, обнаружения и предупреждения вторжений, управления рисками информационной безопасности, выявления уязвимостей и др.

Кроме указанных примеров теоретических изысканий, в открытых источниках можно встретить упоминание практических реализаций. Например, разработанная лабораторией GoogleX система распознавания Vicarious, решающая задачу прохождения CAPTCHA-теста. Исследователям удалось достичь 90 % точности распознавания CAPTCHA от Google, Yahoo, PayPal, Captcha.com и других проектов. Это исследование показывает, что современные CAPTCHA уже не эффективны в качестве теста Тьюринга. В рамках другого проекта GoogleX была создана экспериментальная компьютерная НС, способная самостоятельно распознавать в видеопотоке морфологические объекты, например, кошек. Еще одним примером работающего проекта является работы ряда фирм, ориентированные на борьбу с мошенничеством в банковской сфере.

Распознавание речи – одна из наиболее популярных областей применений НС. Демонстрационная система для дикторонезависимого речевого управления встроенным калькулятором Windows (разработчик – российская компания Нейропроект) способна распознавать 36 команд, сказанных в стандартный микрофон независимо от особенностей голоса и произношения. Использование нейросетевых технологий в криптографии также представлено интересными разработками. Чаще всего применение НС связано c криптосистемами с открытым ключом Меркла-Хэллмана.

Компанией Fein-Marquart Associates Inc разработана программа распознавания почтовых индексов с автоматической дальнейшей сортировкой. Система распознает как типографские, так и написанные от руки цифры. Точность распознавания оценивается значением 98 %. Компания AT&T Bell Laboratories добилась показателей в 0,14 % ошибок НС в процессе распознавания символов при предъявлении обучающей пары из набора представительских выборок, использованной при обучении, и 5,0 % – при распознавании «новых» символов.

## 1.4 Библиотеки и фреймворки для нейронных сетей на языке Python

Сейчас существует большое количество готовых реализаций библиотек для создания и обучения нейронных сетей. Поэтому нет необходимости реализовывать нейронную сеть самостоятельно, можно воспользоваться существующими библиотеками.

1.4.1 Keras – это высокоуровневый API для создания моделей глубокого обучения. Он используется для быстрого создания прототипов, сложных исследований, а также для создания приложений. Три ключевых преимущества Keras API[12]:

* простота в использовании. Keras имеет простой интерфейс, оптимизированный для большинства распространенных задач глубокого обучения. Также он дает конкретные подсказки, как быстро исправить возможные ошибки;
* модульность. Модели Keras строятся при помощи объединения нескольких простых модулей, каждый из которых может быть настроен независимо, и не накладывает каких либо значительных ограничений;
* легко расширить модель. Библиотека позволяет создавать свои собственные модули, чтобы свободно выражать свои идеи для исследования.

1.4.2 Tensorflow – довольно молодой фреймворк для глубокого машинного обучения, разрабатываемый в Google Brain. Долгое время фреймворк разрабатывался в закрытом режиме под названием DistBelief, но после глобального рефакторинга 9 ноября 2015 года был выпущен в открытом доступе [12].

1.4.3 Torch  –  это фреймворк с открытым исходным кодом, предназначенный прежде всего для выполнения большого количества числовых операций. Платформа предлагает множество алгоритмов для быстрого развития сетей глубокого обучения [13].

Данный фреймворк нашел применение в лабораториях искусственного интеллекта Facebook и Twitter. Отличительные особенности Torch:

* фреймворк включает себя множество подпрограмм для индексирования, фрагментирования, транспонирования с N-мерной моделью массива;
* помимо этого, платформа включает в себя процедуры оптимизации, в основном для числовых операций, основанных на нейронных сетях
* платформа поддерживает GPU;
* фреймворк легко взаимодействует с iOS и Android.

1.4.4 Фреймворк Spark MLib от Apache поддерживает библиотеки R, Scala, Java и Python. Его можно загрузить с помощью платформы Hadoop для предоставления алгоритмов машинного обучения, таких как классификация, регрессия и кластеризация.

Помимо платформы Hadoop, его можно загрузить через облачные системы Apache или даже через автономные системы. Отличительные особенности Spark MLib:

* высокая производительность платформы является одним из ключевых ее элементов. Считается, что она в сотни раз быстрее MapReduce от Google;
* Spark  – универсален и способен работать в нескольких вычислительных средах.

1.4.5 H2O – предоставляет доступ к алгоритмам машинного обучения и работает с распространенными средами разработки (Python, Java, Scala, R), системами big data (Hadoop, Spark) и источниками данных (HDFS, S3, SQL, NoSQL). H2O - это end-to-end решение для сбора данных, построения моделей и прогнозов [14].

1.4.6 Theano – Python-библиотека, которая позволяет вам определять, оптимизировать и оценивать математические выражения, особенно с многомерными массивами. Theano позволяет достичь скоростей, близких к решениям на C++ для проблем, связанных с большими объемами данных.

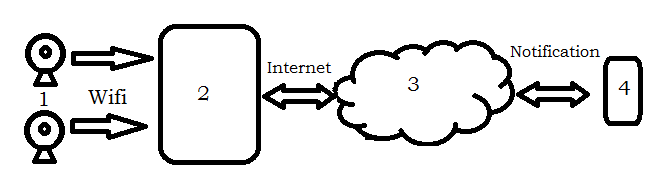
1.4.7 Scikit-learn – использует Python для математической и научной работы. scikit-learn включает инструменты для многих стандартных задач машинного обучения (таких как кластеризация, классификация, регрессия и т. д.). И, так как scikit-learn разрабатывается большим сообществом разработчиков и экспертов по машинным обучению, перспективные новые методы, как правило, включаются в довольно короткий срок.

1.4.8 Shogun – одна из самых старых библиотек машинного обучения. Она постоянно обновляется и может работать на таких языках и средах, как Java, Python, C#, Ruby, R, Lua, Octave и Matlab.

1.4.9 Neon – open source-проект компании Nervana, которая входит в состав Intel. Neon написан преимущественно на Python, а многие стандартные модели нейронных сетей доступны как готовые модели.

## 1.5 Формулирование технического задания на разработку системы

Для создания автоматического контроля комплекса необходимо, чтобы система обладала архитектурой, изображенной на рисунке 1.7



1 – веб-камеры, 2 – ПК, реализованный на Raspberry PI, 3 – глобальная сеть,

4 – мобильное устройство

Рисунок 1.7 – Архитектура системы контроля установки

Изображение с веб-камер с разрешением от 2 МП передается по Wifi в одноплатный компьютер, RaspberryPI. Далее идет обработка изображений с помощью предварительной обученной НС и передача изображений и информаций от НС в Web-приложение и облачное хранилище, в Web-приложении запущено отдельное задание, выполняющее нотификацию (уведомление) с помощью SMS и сообщений в мессенджерах (таких, как WhatsApp, Telegram, Skype) в случае пропадания высокого напряжения на выходе источника питания детектора или при перегрузке, а также при неоптимальных значениях величины давления в криостате. Дополнительной функцией является предоставление доступа к трансляции видео от камер наблюдения.

При разработке НС следует учитывать, что положение камеры может измениться, а также может произойти изменение положения самого объекта (криостат), поэтому обработка изображений должна проходить в 2 этапа:

* на первом этапе изображение передается НС для определения местоположения областей интереса (ИПТ и манометра);
* на втором этапе будет исследоваться сам объект.

Таким образом, мы существенно упрощаем структуры НС.

# 2 Нейронная сеть для контроля работы источника питания детекторов

Питание детектора спектрометрического комплекса может осуществляться разными модулями, в настоящее время в лаборатории мессбауэровской спектроскопии используется источник МКП-011 (фотография этого источника приведена на рисунке. 2.1), который подключается на магистраль интерфейса КАМАК [15] спектрометра СМ-2201 [16]. В процессе обслуживания установки возможны случаи замена модулей и положение источника питания в крейте КАМАК может быть изменено, поэтому нейросеть для определения работы данного блока должна использоваться в паре совместно с нейронной сетью, позволяющей определить положение объекта.

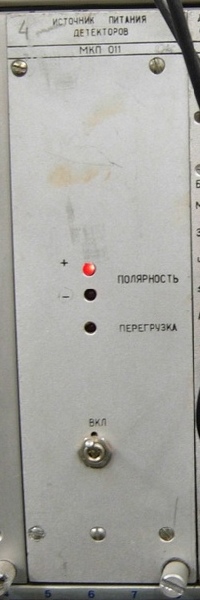


Рисунок 2.1 – Фотография передней панели источник питания МКП 011

Источник питания имеет возможность выдавать высокое напряжение как положительной, так и отрицательной полярности, о выборе режима свидетельствует соответствующий светодиод, дополнительно имеется индикация перегрузки. В случае аварийной ситуации (например, пропадание напряжения в сети 220 В или при грозах) все светодиоды перестают гореть.

При анализе изображений будем выделять следующие ситуации:

– нормальный режим работы (горит светодиод либо отрицательной, либо положительной полярности);

– перегрузка (ситуация, при которой требуется вмешательство со стороны оператора);

– пропадание высокого напряжения в случае аварии (ситуация, при которой требуется вмешательство со стороны оператора).

В подразделе 1.4 были описаны библиотеки, которые позволяют упростить построение и обучение нейросетей. Проанализировав известные библиотеки было принято решение строить нейросеть с использованием библиотек Keras и Tensorflow.

Задача определения состояния источника питания МКП 011 будет решаться как задача классификации объектов. Для этой задачи будет 4 класса объектов (2 класса будут соответствовать рабочему состоянию источника питания, а 2 класса будут соответствовать состояниям, при которых требуется вмешательство со стороны оператора). Основной особенностью настоящей работы является наличие сравнительно малого (всего 400 изображений) набора тренировочных данных, т.к. данная задач в корне отличается от типовых задач определения класса объектов, например самолет или лодка, кошка или собака и т.п. Исходя из особенности (малого числа данных для обучения) было принято решение построить нейросеть по собственной структуре, но первоначально были проведены попытки решения этой задачи с использованием нейросетей VGG-16 и VGG-19 и различных модификаций на их основе. Вероятно, это связанно с огромным числом параметров, которые получаются при генерации модели для таких сетей, например, в одном из вариантов для изображений размером 128х128, в процессе генерации число параметров варьировалось в диапазоне 12-16 млн. Очевидно, что при небольшом числе изображений для обучения и незначительных отличиях в классах изображений (отличия в том, горит светодиод или нет) нейронные сети, построенные на основе VGG-16, VGG-19 и аналогичных работают неэффективно. Поэтому эмпирически была подобрана нейросеть следующей структуры:

* + - 1. cлой свертки областей изображения размером 3 на 3 пиксела, с формированием 64 признаков (фильтров) на выходе слоя, функция активации – relu (см. подраздел 1.1);
      2. слой аналогичный предыдущему – свертка 3 на 3 пиксела с формированием 64 признаков, функция активации – relu;
      3. слой выборки, уменьшает объема входных данных в 4 раза (в 2 раза по вертикали и в 2 раза по горизонтали);
      4. слой случайного отключения нейронов с вероятностью 0,5;
      5. слой для превращения многомерного тензора признаков в одномерную матрицу-столбец;
      6. слой для выбора одного из наиболее подходящих вариантов классификации.

Предложенная нейронная сеть была реализована с использованием Python и библиотек Keras и TensorFlow:

# Подключаем нужные элементы из библиотек

from tensorflow.python.keras.preprocessing.image import ImageDataGenerator

from tensorflow.python.keras.models import Sequential

from tensorflow.python.keras.layers import Conv2D, MaxPooling2D

from tensorflow.python.keras.layers import Activation, Dropout, Flatten, Dense

from tensorflow.python.keras.models import load\_model

# Каталог с данными для проверки

train\_dir = 'pressure\_gauge/шкаф/train'

val\_dir = 'pressure\_gauge/шкаф/val'

# Задаем размеры изображения

img\_width, img\_height = 100, 300

# Размерность тензора на основе изображения для входных данных в нейронную сеть backend Tensorflow, channels\_last

input\_shape = (img\_width, img\_height, 3)

# Количество эпох

epochs = 10

# Размер мини-выборки

batch\_size = 10

# Количество изображений для обучения

nb\_train\_samples = 400

# Количество изображений для проверки

nb\_validation\_samples = 4

# Структура нейронной сети

model = Sequential()

model.add(Conv2D(64, (3, 3),

padding='same',

input\_shape=input\_shape, activation='relu'))

model.add(Conv2D(64, (3, 3), activation='relu',

padding='same'))

model.add(MaxPooling2D(pool\_size=(2, 2)))

model.add(Dropout(0.50))

model.add(Flatten())

model.add(Dense(4))

model.add(Activation('softmax'))

model.summary()

Структуру получившийся НС можно посмотреть в приложении Б.

Далее компилируем модель НС, формируем тренировочные наборы данных и запускаем обучение:

model.compile(loss='categorical\_crossentropy', optimizer='SGD', metrics=['accuracy'])

datagen = ImageDataGenerator(rescale=1. / 255)

train\_generator = datagen.flow\_from\_directory(

train\_dir,

target\_size=(img\_width, img\_height),

batch\_size=batch\_size,

class\_mode='categorical')

val\_generator = datagen.flow\_from\_directory(

val\_dir,

target\_size=(img\_width, img\_height),

batch\_size=batch\_size,

class\_mode='categorical')

model.fit\_generator(

train\_generator,

steps\_per\_epoch=nb\_train\_samples // batch\_size,

epochs=epochs)

Итоговый результат обучения представлен на рисунке 2.2.



Рисунок 2.2 – Результат обучения НС

Зависимости качества обучения нейросети и функции потерь нейросети приведены на рисунке 2.3.

Рисунок 2.3 – Зависимости качества обучения и функции потерь нейросети

Всего за время обучения нейросети прошло 10 эпох. Как видно из рисунка 2.3, нейросеть полностью обучилась за 3 эпохи, а дальнейшие пробеги лишь уменьшали функцию потерь

Результат обучения нейросети был сохранен в виде JSON-модели и файла весов нейронов, в дальнейшем при запуске анализа изображений нейронная сеть не будет обучаться повторно, т.к. будет происходить загрузка модели с весами.

Таким образом, можно сделать вывод: на тренировочном наборе данных получилось добиться точности определения состояния источника питания детекторов 100 % на малом объеме тренировочных данных (400 изображений) при использовании нейронной сети со структурой, состоящей из 6 слоев. При этом данный результат с одной стороны может свидетельствовать о переобучении нейронной сети, но в данном случае это не является проблемой, т.к. под каждый конкретный тип прибора со своими органами управления будет создаваться отдельная нейронная сеть.

# 3 Нейронная сеть для определения показаний давления

Контроль температуры образцов в спектрометрическом комплексе может осуществляться разными криостатами, в настоящее время в лаборатории мессбауэровской спектроскопии используется криостат Oxford Instruments с установленным манометром ДМ 02-063-1-М (фотография этого манометра приведена на рисунке 3.1). Манометр измеряет давление в системе от 0 до 1,6 кгс/см2 с шагом в 0,05. В соответствии с этим выделим 33 возможных значения, а промежуточные значения будем округлять. В процессе обслуживания установки при замене сосуда Дьюара возможно изменение положения криостата, поэтому НС для определения показаний манометра должна использоваться в паре совместно с нейронной сетью, позволяющей определить положение объекта.



Рисунок 3.1 – Манометр ДМ02-063-1-М

Задача определения показаний манометра ДМ02-063-1-М будет решаться как задача классификации объектов. Для этой задачи будет 33 класса объектов (по 1 классу на одно деление измерительной шкалы). Для тренировки НС средствами графического редактора Photoshop и библиотеками по работе с изображениями языка программирования Python был подготовлен набор тренировочных данных в 11880 фотографий (в каждом классе находиться 360 изображений манометра с одним значением, отличающиеся между собой углом поворота изображения). Исходя из особенности (в общем доступе нет информации об использовании НС для снятия показаний с измерительной шкалы) было принято решение построить нейросеть по собственной структуре, но первоначально были проведены попытки решения этой задачи с использованием нейросетей VGG-16 и VGG-19 и различных модификаций на их основе. При обучении таких сетей на задачу распознавания показаний процент правильности ответов не возрастал. Возможно это было связано с тем, что в процессе генерации число параметров варьировалось в диапазоне 16-90 млн.. Поэтому эмпирически была подобрана нейросеть следующей структуры:

* + - 1. слой свертки областей изображения размером 3 на 3 пикселя, с формированием 32 признаков (фильтров) на выходе слоя, функция активации – relu (см. подраздел 1.1);
      2. слой аналогичный предыдущему – свертка 3 на 3 пикселя с формированием 32 признаков, функция активации – relu;
      3. слой выборки, уменьшает объема входных данных в 4 раза (в 2 раза по вертикали и в 2 раза по горизонтали);
      4. слой случайного отключения нейронов с вероятностью 0,5;
      5. слой свертка 3 на 3 пикселя с формированием 64 признаков, функция активации – relu;
      6. слой – свертка 3 на 3 пикселя с формированием 64 признаков, функция активации – relu;
      7. слой выборки, уменьшает объема входных данных в 4 раза;
      8. слой случайного отключения нейронов с вероятностью 0,5;
      9. слой – свертка 3 на 3 пикселя с формированием 128 признаков, функция активации – relu;
      10. слой - свертка 3 на 3 пикселя с формированием 128 признаков, функция активации – relu;
      11. слой – свертка 3 на 3 пикселя с формированием 128 признаков, функция активации – relu;
      12. слой выборки, уменьшает объема входных данных в 4 раза;
      13. слой случайного отключения нейронов с вероятностью 0,5;
      14. слой для превращения многомерного тензора признаков в одномерную матрицу-столбец;
      15. слой для выбора одного из наиболее подходящих вариантов классификации.

Ниже приведен листинг программы:

# Подключаем нужные элементы из библиотек

from tensorflow.python.keras.preprocessing.image import ImageDataGenerator

from tensorflow.python.keras.models import Sequential

from tensorflow.python.keras.layers import Conv2D, MaxPooling2D

from tensorflow.python.keras.layers import Activation, Dropout, Flatten, Dense

# Каталог с данными для обучения

train\_dir = '/pressure\_gauge/780x780/train'

# Каталог с данными для проверки

val\_dir = '/pressure\_gauge/780x780/val'

# Каталог с данными для тестирования

test\_dir = '/pressure\_gauge/780x780/test'

# Размеры изображения

img\_width, img\_height = 150, 150

# Размерность тензора на основе изображения для входных данных в нейронную сеть backend Tensorflow, channels\_last

input\_shape = (img\_width, img\_height, 3)

# Количество эпох

epochs = 30

# Размер мини-выборки

batch\_size = 40

# Количество изображений для обучения

nb\_train\_samples = 11880

# Количество изображений для проверки

nb\_validation\_samples = 1815

# Количество изображений для тестирования

nb\_test\_samples = 1792

# Создаем структуру НС

model = Sequential()

model.add(Conv2D(32, (3, 3), padding='same',input\_shape=input\_shape, activation='relu'))

model.add(Conv2D(32, (3, 3), activation='relu', padding='same'))

model.add(MaxPooling2D(pool\_size=(2, 2)))

model.add(Dropout(0.5))

model.add(Conv2D(64, (3, 3), padding='same', activation='relu'))

model.add(Conv2D(64, (3, 3),padding='same', activation='relu'))

model.add(MaxPooling2D(pool\_size=(2, 2)))

model.add(Dropout(0.5))

model.add(Conv2D(128, (3, 3), padding='same', activation='relu'))

model.add(Conv2D(128, (3, 3), padding='same', activation='relu'))

model.add(Conv2D(128, (3, 3),padding='same', activation='relu'))

model.add(MaxPooling2D(pool\_size=(2, 2)))

model.add(Dropout(0.5))

model.add(Flatten())

model.add(Dense(33))

model.add(Activation('softmax'))

model.summary()

Структуру получившейся НС можно посмотреть в приложении B.

Далее компилируем модель НС, формируем тренировочные наборы данных и запускаем обучение:

model.compile(loss='categorical\_crossentropy',

optimizer='SGD',

metrics=['accuracy'])

datagen = ImageDataGenerator(rescale=1. / 255)

train\_generator = datagen.flow\_from\_directory(

train\_dir,

target\_size=(img\_width, img\_height),

batch\_size=40,

class\_mode='categorical')

val\_generator = datagen.flow\_from\_directory(

val\_dir,

target\_size=(img\_width, img\_height),

batch\_size=2,

class\_mode='categorical')

test\_generator = datagen.flow\_from\_directory(

test\_dir,

target\_size=(img\_width, img\_height),

batch\_size=1,

class\_mode='categorical')

model.fit\_generator(

train\_generator,

steps\_per\_epoch= 297,

epochs=epochs,

validation\_data=val\_generator,

validation\_steps=99)

Итоговый результат обучения представлен на рисунке 3.2.



Рисунок 3.2 – Результат обучения НС

Зависимости качества обучения нейросети приведены на рисунке 3.3.

Рисунок 3.3 – Зависимости качества обучения и функции потерь нейросети

Всего за время обучения нейросети прошло 50 эпох. Как видно из рисунка 3.3, скорость обучения НС упала после 40 эпохи. Это связано с тем, что изначально проводилось обучение в 40 эпох, а после проходило дообучение сети.

Результат обучения нейросети был сохранен в виде JSON-модели и файла весов нейронов, в дальнейшем при запуске анализа изображений нейронная сеть не будет обучаться повторно, т.к. будет происходить загрузка модели с весами.

Вывод: в ходе обучения сети получилось достигнуть точности на тренировочном наборе данных в 98 %. На дополнительном наборе – 81 %. Сложность в достижении 100 % определения показаний заключается в большом количестве возможных показаний и малом количестве отличий между фотографиями. Для дообучения сети следует подготовить дополнительный набор данных.

# 4 Нейронная сеть для определения местоположений областей интереса на изображении

Для создания НС для определения местоположения ИПД и манометра на изображении с видеокамер следует использовать библиотеку TensorFlow Object Detection API [17]. Для начала стоит подготовить набор данных для обучения. Для этого стоит сделать аннотации к более 200 фотографиям (по 100 на каждый объект). Это можно проводить при помощи пакета LabelImg.

После создания фотографий и их аннотаций следует преобразовать аннотации в так называемый формат TFRecord. Это происходит в 2 шага:

1. Преобразование отдельных файлов .xml в единый файл .csv для каждого набора данных.
2. Преобразование файлов .csv каждого набора данных в файлы .record (формат TFRecord).

Листинг программ для преобразования представлен в приложениях Г и Д.

Следующим шагом следует начать обучение НС. Нейросеть ssd\_resnet\_50\_fpn\_coco обеспечивает наилучшую точность определения относительно производительности (согласно ТЗ время обработки требуется менее 200 мс). Для ее обучения следует выполнить код из приложения Е.

Результатом выполнения работы является НС, обученная на распознавание ИПД и манометра криостата.

# 5 Безопасность жизнедеятельности

В нашей стране разработана комплексная система законов и стандартов, определяющих безопасность человека и охрану окружающей среды. Во время прохождения преддипломной практики необходимо соблюдение этих законов, а так же систему стандартов безопасности труда (ССБТ) [18], санитарные нормы (СН), строительные нормы и правила (СНиП), гигиенические нормы (ГН), санитарные правила и нормы (СанПиН). Соблюдение этих норм и свода федеральных законов обеспечит безопасность условий прохождения преддипломной практики на месте дипломирования.

## 5.1 Описание объекта дипломирования

Тема дипломной работы – система машинного зрения для контроля мессбауэровского спектрометрического комплекса.

Дипломная работа выполнялась в лаборатории мессбауровской спектроскопии на кафедре экспериментальной физики в составе Уральского Федерального Университета имени Б.Н. Ельцина. Санитарно-защитная зона не предусмотрена. Лаборатория находится на территории города Екатеринбурга, в непосредственной близости от учебных корпусов расположены жилые дома. Для города характерен умеренно-континентальный климат с резким изменением погодных условий и ярко выраженными сезонами года.

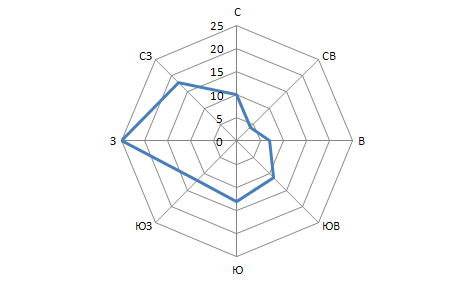


Рисунок 5.1 - Роза ветров [19]

Среднегодовая скорость ветра – 3,4 м/с, среднегодовая температура +2,2 °C, средняя температура января −14,2 °C, средняя температура июля +16,4 °C, среднегодовая влажность воздуха – 74,8 %, среднегодовое количество осадков – 497 мм [19].

## 5.1.1 Описание рабочего места

Выполнение дипломной работы проходило в лаборатории мессбауровской спектроскопии, расположенной в здании Физико-технологического института (см. Рисунок 5.5.2). Помещение, в котором проводились исследования, имеет следующие размеры: длина A=5 м, ширина B=4,5 м, высота H=2,5 м. Площадь и объём рабочего помещения составляют:

м2, (5.1)

(5.2)

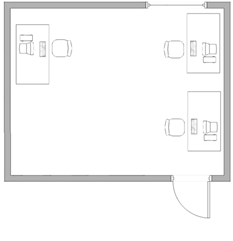


Рисунок 5.2 - Схема помещения в лаборатории

В помещении находится 3 рабочих места, на которых находятся 3 ПК. Таким образом, на одного сотрудника приходится площадь 7,5 м2 и объём 18,75 м3. Согласно СанПиН 2.2.2/2.4.1340-03 [20] площадь на одно рабочее место с ПК c ВДТ на базе плоских дискретных экранов (жидкокристаллические, плазменные) для взрослых пользователей должна составлять не менее 4,5 м2, а объем не менее 18,0 м3. Следовательно, можно сделать вывод, что помещение соответствует данным требованиям для эксплуатации.

## 5.1.2 Интенсивность труда

Класс условий труда - 2. Большая часть всех производимых работ производится сидя и сопровождается легким физическим напряжением, поэтому интенсивность труда относится к легким работам категории Ia - работы с интенсивностью энергозатрат до 139 ккал/ч.

## 5.1.3 Микроклимат

Микроклимат в помещении определяется температурой, влажностью и скоростью движения воздуха, а также температурой окружающих поверхностей.

Нормируемые параметры микроклимата устанавливаются в соответствии с требованиями ГОСТ 12.1.005-88.

В СНиП 41-01-2003 [21] установлены следующие оптимальные величины показателей микроклимата (см. Таблица 5.1) на рабочих местах производственных помещений для категории работ по уровню энергозатрат:

Таблица 5.1 - Показатели микроклимата

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Параметры микроклимата | Значения параметров | | | |
| Теплый период года | | Холодный период года | |
| Нормативное значение | Факт | Нормативное значение | Факт |
| Температура воздуха, °C | 23-25 | 23 | 22-24 | 24 |
| Температура поверхностей, °C | 21-25 | 22 | 22-26 | 22 |
| Относительная влажность воздуха, % | 40-60 | 55 | 40-60 | 52 |
| Скорость движения воздуха, м/с | 0,1 | 0,1 | 0,1 | 0,1 |

Средняя температура воздуха в рабочей комнате составляет плюс 23 °С, относительная влажность – 55 %.

Микроклимат в помещении рабочей комнаты соответствует оптимальным значениям по СанПиН 2.2.2/2.4.1340-03. Произведем расчет кратности обмена воздуха в помещении, по формуле (5.3):

, (5.3)

где Vвент - объем воздуха, необходимый для обмена, м3;

- объем рабочего помещения, м3.

Объем рабочего помещения составляет – 56,25 м3.

Объем воздуха Vвент определим из уравнения теплового баланса, по формуле (5.4) :

, (5.4)

где Qизбыт - избыточная теплота, примем равной 350 Вт;

С - удельная теплоемкость воздуха, 1006 Дж/(кг∙К) ;

Y - плотность воздуха, 1,225 кг/м3;

tуход – температура уходящего воздуха, 23 оС;

tприход – температура приходящего воздуха, 19 оС.

Кратность воздухообмена составит:

(5.5)

Оптимальным вариантом является кондиционирование воздуха, т.е. автоматическое поддержание его состояния в рабочем помещении в соответствии с определенными требованиями независимо от изменения состояния наружного воздуха и условий в самом помещении.

Человек начинает ощущать движение воздуха при его скорости 0,1 м/с. Поэтому норма скорости, установленная для оптимальных значений микроклимата является 0,1 м/с.

## 5.1.4 Освещенность

Для искусственного освещения используются люминесцентные лампы, у которых высокая световая отдача (до 75 лм/Вт и более), малая яркость светящейся поверхности, близкий к естественному спектральный состав излучаемого света, что обеспечивает хорошую цветопередачу.

Все методы расчета искусственного освещения основаны на формулах, связывающих освещенность с характеристиками ламп.

Рассчитаем искусственную освещенность по формуле (5.6), учитывая, что в помещении имеется 18 люминесцентные лампы мощностью по 18 Вт

, (5.6)

где ФК – световой поток одной лампы, ФК = 1350 лм;

N – число ламп в помещении, N = 14;

ηи – коэффициент затенения, ηи = 0,6;

Кз – коэффициент запаса, Кз = 1,2;

Sn – площадь освещаемого помещения, Sn = 22,5 м2;

z – коэффициент неравномерности освещения, z = 0.9.

Подставив значения в формулу (5.6) получим, что минимальная освещенность составляет 466 лк. Согласно значениям, приведенным в СанПин 2.2.2/1340-03 освещённость на поверхности стола в зоне размещения рабочего документа должна быть 300 - 500 лк. следовательно освещение помещения удовлетворяет требуемым значениям.

## 5.1.5 Эргономичность

Конструкция рабочего места и взаимное расположение всех его элементов (сиденье, органы управления, средства отображения информации) соответствуют антропометрическим, физиологическим, психологическим, гигиеническим и социально-психологическим требованиям. Выполнение трудовых операций происходит в пределах зоны моторного поля. Монитор находится на расстоянии 70 см от глаз трудящегося, что является допустимым показателем.

Для обеспечения комфортной работы, рабочие места в рабочей комнате расположены на расстоянии 215 см. Высота рабочей поверхности стола составляет – 760 мм, габаритные размеры рабочей области стола –   
1400х1200 мм, все другие антропометрические показатели по организации рабочего места соответствуют требованиям СанПиН 2.2.2/1340-03.

## 5.2 Опасные производственные факторы

Опасные производственные факторы – факторы, воздействие которых на работающего в определённых условиях могут привести к травме, внезапному резкому ухудшению здоровья или смерти.

Основным опасным производственным фактором на данном рабочем месте при проведении испытаний и проектировании является электрический ток.

## 5.2.1 Электробезопасность

Под определением электробезопасности понимают систему организационных мероприятий и технических средств, позволяющих предотвратить вредное и опасное воздействие электрического тока, электрической дуги, электромагнитного поля и статического электричества.

С точки зрения опасности поражения человека электрическим током лаборатория относится к помещениям с повышенной опасностью в соответствие с ГОСТ 12.1.019-2009.

Электропитание большей части аппаратуры обеспечивается от сети переменного тока напряжением 220 В. ГОСТ 12.1.038-82 устанавливает предельно допустимые уровни напряжений и токов при аварийных режимах работы.

В помещении лаборатории приняты следующие меры по защите от поражения электрическим током:

* проложены шины зануляющего контура и имеются автоматы для защиты от токов короткого замыкания;
* силовая электрическая сеть проложена в металлических трубах, которые сварным соединением связаны с контуром зануления.

Общая схема защитного зануления установки представлена на рисунке 5.3.

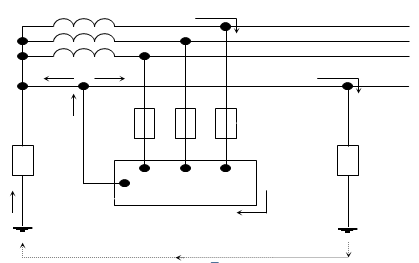


Рисунок 5.3 – Cхема защитного зануления

## 5.2.2 Вредные факторы и мероприятия по их устранению

Вредные производственные факторы – факторы, воздействие которых на работающего человека в определенных условиях приводит к заболеванию, снижению работоспособности или профессиональному заболеванию.

При работе в лаборатории волоконной оптики основными вредными факторами являются: шум, запылённость. Влияние электромагнитного излучения можно не учитывать, так как вся современная выпускаемая техника проходит обязательную сертификацию на безопасность человека и величина создаваемого электромагнитного излучения ничтожно мала. Источники ощутимой для человека вибрации также отсутствуют.

Шум в рабочем помещении может оказывать вредное влияние как на органы слуха, так и нервную систему, и внутренние органы. В этой связи необходим контроль за уровнем шума в помещениях, где работают люди. В помещении, где проводилась работа основным источником шума являются вентиляторы, уровень шума от которых не превышает 40 дБа, таким образом, в согласии с ГОСТ 12.1.003-83, данное помещение удовлетворяет требованиям шумоизоляции.

## 5.2.3 Пожарная охрана

Под пожарной охраной понимают систему государственных и общественных мероприятий, направленных на охрану от огня людей и собственности.

Горение – физико-химическая реакция, сопровождающаяся выделением тепла, света.

Пожар – неконтролируемый процесс горения, причиняющий материальный ущерб, вред жизни и здоровью людей, интересам общества и государства.

Помещение, в котором проходило выполнение дипломной работы, согласно НПБ 105-95 по категории взрывопожарной опасности относится к категории В и характеризуется наличием в помещении только несгораемых веществ и материалов в холодном состоянии.

В помещении предусмотрена противопожарная защита в виде автоматической пожарной сигнализации, а также наличием средств первичного пожаротушения. Стены и перекрытия помещения выполнены из бетона и относятся к несгораемым.

Предотвращение пожара в помещении достигается минимальным количеством предметов из горючих материалов, их безопасным расположением, а также отсутствием легко воспламеняющихся веществ. При возникновении пожарной ситуации все сотрудники, находящиеся в лаборатории организованно должны покинуть помещение согласно специально разработанному плану эвакуации, находящемуся на каждом этаже здания.

Схема эвакуации электротехнической лаборатории представлена на рисунке 5.4.

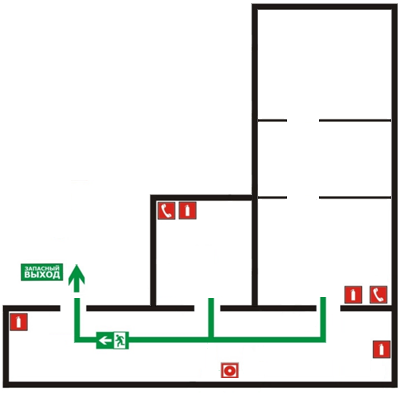


Рисунок 5.4 - План эвакуации персонала из лаборатории мессбауэровской спектроскопии

## 5.2.4 Возникновение ЧС

Чрезвычайная ситуация (ЧС) – обстановка, сложившаяся на определенной территории в результате катастрофы, аварии, стихийного бедствия, которая может повлечь за собой (или повлекла) человеческие жертвы, ущерб здоровью людей и окружающей среде, значительные материальные потери, а также нарушение условий жизнедеятельности.

За проведением защитных мероприятий, подготовкой всех необходимых документов и инвентаря, обучением персонала, его своевременным информированием в УрФУ следит специальный орган – штаб ГО и ЧС. Именно на него возложено и руководство действиями персонала и проведение спасательных работ при возникновении ЧС.

На случай возникновения различного рода ЧС техногенного или природного характера в УрФУ разработаны различные планы оперативных действий, направленных на минимизацию наносимого ЧС урона и спасение жизни и здоровья персонала, и студентов.

В состав УрФУ входят ряд подразделений, которые занимаются работой с источниками излучений. Хотя используемые образцы не обладают большой активностью, но при неправильном обращении с ними возможны возникновения ЧС. Также необходимо учитывать наличие в городе Екатеринбурге и соседних областях крупных предприятий, использующих в производственном цикле достаточно мощные источники излучений, и то, что наша область является транзитной при перевозке радиоактивных отходов на переработку. Все это заставляет уделять значительно внимание вопросам радиационной защиты.

Радиационная защита – комплекс мероприятий, направленный на защиту живых организмов от ионизирующего излучения, а также, изыскание способов ослабления поражающего действия ионизирующих излучений.

Заблаговременно проводятся следующие защитные мероприятия:

* разрабатываются планы действий на случай возникновения ЧС, связанных с радиационными авариями;
* разрабатываются и внедряются системы контроля за радиационной обстановкой на объектах, работающих с источниками излучения;
* определяется степень опасности данных объектов и требуемые защитные меры;
* поддерживается требуемый запас средств индивидуальной защиты, медикаментов и дезактивации;
* создаются специальные укрытия и защитные сооружения;
* проводится своевременная подготовка персонала к действиям при подобных ЧС.

На Урале сильные грозы не являются редким явлением, поэтому в УрФУ предусмотрена система защиты от попадания молний, которое может приводить к пожарам, поражению людей и выходу из строя различного оборудования. Для этого специальный заземлитель – громоотвод. Он состоит из трех основных частей: токоотвода, молниеприемника и контура заземления.

## 5.2.5 Молниезащита

Молниеприемник – обычный металлический проводник, устанавливаемый вертикально на крыше здания так, чтобы его конец был самой высокой точкой здания. Изготавливается из малоокисляющихся на воздухе сплавов.

Токоотвод – обычный провод, соединяющий молниеприемник и систему заземления.

Контур заземления – заземлитель, состоящий из нескольких электродов, соединенных друг с другом и смонтированных вокруг здания по его контуру. В случае, если силуэт заземления уже имеется, правила требуют, чтобы заземление молниезащиты и электрических приборов было единым.

## 5.3 Выводы

В соответствии с принятыми нормами в лаборатории мессбауэровской спектроскопии обеспечивается необходимый микроклимат, минимальный уровень шума, созданы удобные и правильные с точки зрения эргономики рабочие места, соблюдены требования по освещенности и требования к ЭВМ. Специальные мероприятия обеспечивают электробезопасность и пожаробезопасность сотрудников. В целом условия труда в лаборатории волоконной оптики соответствуют общепринятым нормам, сотрудникам обеспечены комфорт и благоприятные условия труда.

# 6 Природопользование и охрана окружающей среды

## 6.1 Введение

В настоящее время проблема роста использования природных ресурсов все чаще возникает в обществе. Рост интереса к данному объекту можно объяснить развитием как отечественной, так и мировой экономики, а также ростом НТП. Проблема рационального природопользования с каждым днем становится более актуальной. В связи с этим перед человеком ставятся новые задачи: максимально возможное эффективное использование ресурсов, обеспечение безопасности окружающей среды, а также ее охрана.

Введем основные термины природопользования.

Природопользование – это качественный и количественный учет природной среды и природных ресурсов с прогнозированием возможностей их общественного, а не только производственного использования.

Охрана окружающей среды – комплекс мер, предназначенных для ограничения отрицательного влияния человеческой деятельности на [окружающую среду](https://ru.wikipedia.org/wiki/%D0%9E%D0%BA%D1%80%D1%83%D0%B6%D0%B0%D1%8E%D1%89%D0%B0%D1%8F_%D1%81%D1%80%D0%B5%D0%B4%D0%B0) ([природу](https://ru.wikipedia.org/wiki/%D0%9F%D1%80%D0%B8%D1%80%D0%BE%D0%B4%D0%B0)) [30].

## 6.2 Влияние объекта дипломной работы на окружающую среду

Дипломная работа выполнялась в лаборатории мессбауровской спектроскопии на кафедре экспериментальной физики в составе Уральского Федерального Университета имени Б.Н. Ельцина. Данное учебное заведение осуществляет учебную и научную деятельность и имеет ряд экспериментальных производств, связанных с воздействием на окружающую среду. Исследование выполнено при помощи персонального компьютера.

Для описанного рабочего места факторами экологического риска являются:

* факторы, обусловленные утилизацией электронной техники;
* физические факторы (электромагнитное и акустическое излучения);
* потребление ресурсов регулирует Федеральный закон "Об энергосбережении и о повышении энергетической эффективности и о внесении изменений в отдельные законодательные акты Российской Федерации" от 23.11.2009 N 261-ФЗ (последняя редакция);
* образование твердых отходов.

При выполнении настоящей дипломной работы использовался персональный компьютер с жидкокристаллическим монитором (ЖК). Особенность данного типа мониторов заключается в том, что они практически не являются источниками каких-либо опасных электромагнитных излучений, в отличие от мониторов на основе электронно-лучевых трубок.

## 6.3 Утилизация электронной техники

Компьютерная техника имеет относительно короткий срок эксплуатации. При утилизации компьютерной техники некоторые ее составляющие могут быть подвергнуты переработке и повторному использованию. В современном мире процесс возвращения отходов в техногенез получил название рециклинг. В частности, различные виды пластика и драгоценных металлов могут использоваться повторно.

Обезвреживание отходов (уменьшение массы отходов, изменение их состава, физико-химических свойств) требуется для отдельных компонентов, таких металлы как свинец, сурьма, ртуть, кадмий, мышьяк, входящие в состав электронных компонентов, трансформируются под воздействием внешних условий в очень токсичные соединения. Грамотное обезвреживание таких веществ с последующей утилизацией – важная проблема экологии.

По итогам анализа факторов воздействия на окружающую среду можно сделать следующие выводы:

* при соблюдении всех норм и правил эксплуатации установок выполнение различных работ в лаборатории безопасно для персонала и не оказывает значительного влияния на экологическую обстановку;
* использование современных ПК с жидкокристаллическими мониторами вместо стандартных ЭЛТ-мониторов позволяет не только снизить уровень потребления энергии, но и минимизировать воздействие вредного электромагнитного излучения;
* утилизация компьютерной техники должна отвечать требованиям по охране окружающей среды и не быть убыточной.

## 6.4 Способы обеспечения экологической безопасности

Выявленными факторами экологического риска являются факторы, обусловленные использованием вредных веществ, утилизацией электронной техники и физические факторы – электромагнитное и акустическое излучение.

Снижение экологической опасности при использовании вредных веществ может быть достигнуто за счет правильной эксплуатации санитарно-технического оборудования и устройств (отопления, вентиляции, водопровода, канализации), специальной подготовки обслуживающего персонала и контроля за содержанием вредных веществ в воздухе рабочей зоны.

Люминесцентные лампы после использования утилизируются и демеркуризируются. Отработавшая техника списывается и передается в административно-хозяйственную часть. Постановлением Правительства Российской Федерации № 681 утверждены правила обращения с отходами производства и потребления в части осветительных устройств, электрических ламп, ненадлежащие сбор, накопление, использование, обезвреживание, транспортирование и размещение которых может повлечь причинение вреда жизни, здоровью граждан, вреда животным, растениям и окружающей среде [31].

## 6.5 Потребление энергоресурсов

Потребление энергоресурсов регулирует 261-ФЗ «Об энергосбережении» [32]. Данный закон распространяемся на деятельность, связанную с использованием энергетических ресурсов.

Основные принципы:

* эффективное рациональное использование энергетических ресурсов;
* поддержка и стимулирование энергосбережения;
* планирование энергосбережения и повышения энергетической эффективности.

Использование энергетически ресурсов с учетом ресурсных, производственно-технологических, экологических и социальных условий.

## 6.6 Утилизация электронной техники и отходов инфраструктуры

Во время жизненного цикла разрабатываемой системы, согласно ФККО-2014 утилизируются следующие отходы:

* использованные книги, журналы, брошюры, проспекты, каталоги – код 4 05 122 01 60 5, класс опасности V;
* отходы бумаги и картона от канцелярской деятельности и делопроизводства – код 4 05 122 02 60 5, класс опасности V;
* лампы ртутные, ртутно-кварцевые, люминесцентные, утратившие потребительские свойства – код 4 71 101 01 52 14, класс опасности I;
* мусор от офисных и бытовых помещений организаций несортированный (исключая крупногабаритный) – код 7 33 100 01 72 4, класс опасности IV;
* системный блок компьютера, утративший потребительские свойства – код 4 81 201 01 52 4, класс опасности IV;
* принтеры, сканеры, многофункциональные устройства (МФУ), утратившие потребительские свойства – код 4 81 202 01 52 4, класс опасности IV;
* клавиатура, манипулятор «мышь» с соединительными проводами, утратившие потребительские свойства – код 4 81 204 01 52 4, класс опасности IV.

Утилизация данных видов отходов осуществляется согласно ФЗ №89 «Об отходах производства и потребления». Настоящий Федеральный закон определяет правовые основы обращения с отходами производства и потребления в целях предотвращения вредного воздействия отходов производства и потребления на здоровье человека и окружающую среду, а также вовлечения таких отходов в хозяйственный оборот в качестве дополнительных источников сырья.

Согласно приказу №511 от 15.06.2001 года Министерства природных ресурсов Российской Федерации на территории всей России установлено 5 классов опасности отходов производства и жизнедеятельности человека, а также степень влияния на окружающую среду и критерии вредного воздействия:

1-й класс – чрезвычайно опасные. Степень вредного воздействия на окружающую среду отходов этого класса характеризуется как «очень высокая». В результате накопления отходов первого класса происходят необратимые нарушения в экологической системе, а период ее восстановления отсутствует.

2-й класс – высокоопасные. Степень вредного воздействия оценивается как «высокая». Экологическое равновесие системы сильно нарушается, а период восстановления системы и ее компонентов составляет не менее 30 лет после полного устранения источника воздействия.

3-й класс – умеренно опасные. Средняя степень вредного воздействия с периодом самовосстановления от 10 лет после снижения уровня воздействия.

4-й класс – малоопасные. Установлена низка степень вредного воздействия на природную среду, а период восстановления составляет от 3-х лет.

5-й класс – практически неопасные. Степень воздействия – очень низкая, экологическая система и ее компоненты не нарушены.

Анализируя факторы экологического риска, которые могут возникнуть при работе в лаборатории математического моделирования, можно сделать следующие выводы.

Использование ПК обуславливает возникновение проблемы, связанной с энергопотреблением и утилизацией компьютерной техники – это ведет к загрязнению окружающей среды. Необходимо построить процесс утилизации отслужившей компьютерной техники, таким образом, чтоб он отвечал требованиям по охране окружающей среды. В ходе выполнения проекта потребление электроэнергии осуществлялось согласно 261-ФЗ, этот пункт не наносит вред окружающей среде.

Согласно ФЗ РФ «Об охране окружающей среды» от 10.01.2002 № 7-ФЗ (последняя редакция), дата обращения 13 мая 2019 года, необходимо соблюдать основные принципы ведения хозяйственной или иной деятельности, оказывающей воздействие на окружающую среду. Проанализировав их, можно сделать вывод о вреде проекта окружающей среде.

Выполнение проекта – часть производственного процесса, поэтому сделаем вывод о соблюдении принципов рационального природопользования при выполнении работы, разработанный проект не наносит вред экологии.

# ЗАКЛЮЧЕНИЕ

В результате выполнения работы была разработана архитектура системы контроля спектрометрического комплекса и выбраны технические решения для ее реализации. Были предложены архитектуры двух нейронных сетей для контроля состояния источника питания детекторов МКП 011 и криостата мессбауэровского спектрометрического комплекса.

На основе предложенных в данной работе архитектур были созданы нейросети для определения показаний на манометре и режима работы ИПД на основе решения задачи классификации объектов, а также определения местоположения этих объектов на изображении с помощью сверточных нейронных сетей. Нейронные сети были созданы с использованием Python, Keras и TensorFlow. Для их обучения были сформированы базы данных, в сумме состоящие из более 20 тысяч фотографий, но, в целом, такой объем изображений, можно считать небольшим. В результаты точность определения значений манометра и состояния ИПД составила 0,9 – 1,0 на тренировочном и тестовом наборе данных. Данные нейронные сети можно адаптировать (путем создания другого набора тренировочных и тестовых данных) для анализа состояния и контроля других научных и промышленных установок. Был предложен алгоритм разработки и обучения НС для распознавания местоположения объектов на изображении.

Дальнейшее развитие данной работы предусматривает:

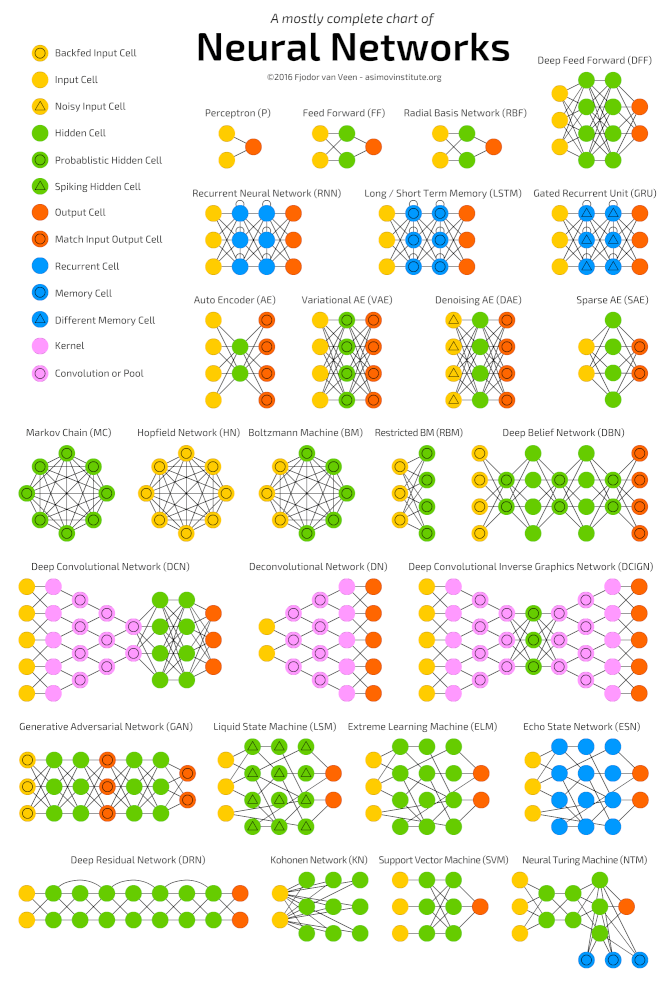
* запуск непрерывной (с частотой 5 кадров/сек) обработки изображений, поступающих от IP-камер;
* запуск Web-приложения для управления процессом анализа изображений, настройкой способов хранения и т.п.;
* запуск процесса для отправки уведомлений (СМС, сообщения в мессенджерах) оператору в случае возникновения аварийной ситуации;
* расширение числа контролируемых параметров и увеличение числа камер.

# СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. A high velocity resolution Mössbauer spectrometric system for biomedical research / V.A. Semionkin, M.I. Oshtrakh, O.B. Milder, E.G. Novikov // Bulletin of the Russian Academy of Science: Physics. – 2010. – V. 74. – P. 416–420.
2. Oshtrakh M.I., Semionkin V.A. Mössbauer spectroscopy with a high velocity resolution: Advances in biomedical, pharmaceutical, cosmochemical and nanotechnological research // Spectrochimica Acta, Part A. – 2013. – V. 100. – P. 78–87.
3. Mössbauer spectroscopy with high velocity resolution: New possibilities of chemical analysis in material science and biomedical research / M.I. Oshtrakh, V.A. Semionkin, V.I. Grokhovsky, O.B. Milder, E.G. Novikov // Journal of Radioanalytical and Nuclear Chemistry. – 2009. – V. 281. – P. 63–67.
4. Szeliski R. Computer Vision: Algorithms and applications. – London : Springer, 2011. – P. 812.
5. Что такое компьютерное зрение? [Электронный ресурс] // RUSBASE. – URL: https://rb.ru/story/chto-takoe-mashinnoe-zrenie/ (дата обращения: 08.03.2019).
6. Горбачевская Е.Н. Классификация нейронных сетей // Вестник ВУиТ. – 2012. – №2. – 19 c.
7. Как работает нейронная сеть [Электронный ресурс] // Neurohive. –URL: <https://neurohive.io/ru/osnovy-data-science/osnovy-nejronnyh-setej-algoritmy-obuchenie-funkcii-aktivacii-i-poteri/>(дата обращения: 08.03.2019).
8. Практика: Распознавание предметов одежды на Keras [Электронный ресурс] // ASozykin–URL: https://www.asozykin.ru/courses/nnpython-lab1 (дата обращения: 08.03.2019).
9. VGG нейронная сеть [Электронный ресурс] // Quora. – URL: <https://www.quora.com/What-is-the-VGG-neural-network>/ (дата обращения: 15.02.2019).
10. Понимание и реализация архитектур ResNet [Электронный ресурс] // Medium. – URL: <https://medium.com/@14prakash/understanding-and-implementing-architectures-of-resnet-and-resnext-for-state-of-the-art-image-cf51669e1624>/ (дата обращения: 15.02.2019).
11. Нейронные сети в промышленности и информационных технологиях [Электронный ресурс] // Ицрон. – URL: <http://izron.ru/articles/razvitie-tekhnicheskikh-nauk-v-sovremennom-mire-sbornik-nauchnykh-trudov-po-itogam-mezhdunarodnoy-na/sektsiya-2-informatika-vychislitelnaya-tekhnika-i-upravlenie-spetsialnost-05-13-00/neyronnye-seti-v-promyshlennosti-i-informatsionnykh-tekhnologiyakh/> (дата обращения: 15.02.2019).
12. Keras [Электронный ресурс] // TensorFlow. – URL: <https://www.tensorflow.org/guide/keras?hl=ru>/ (дата обращения: 15.02.2019).
13. Лучшие фреймворки в мире для искусственного интеллекта [Электронный ресурс] // Medium. – URL: <https://medium.com/nuances-of-programming/%D1%82%D0%BE%D0%BF-9-%D1%84%D1%80%D0%B5%D0%B9%D0%BC%D0%B2%D0%BE%D1%80%D0%BA%D0%BE%D0%B2-%D0%B2-%D0%BC%D0%B8%D1%80%D0%B5-%D0%B8%D1%81%D0%BA%D1%83%D1%81%D1%81%D1%82%D0%B2%D0%B5%D0%BD%D0%BD%D0%BE%D0%B3%D0%BE-%D0%B8%D0%BD%D1%82%D0%B5%D0%BB%D0%BB%D0%B5%D0%BA%D1%82%D0%B0-%D1%87%D0%B0%D1%81%D1%82%D1%8C-2-3bd87aa6ea60>/ (дата обращения: 15.02.2019).
14. Машинное обучение [Электронный ресурс] // Apptractor. – URL: <https://apptractor.ru/ML/> (дата обращения: 15.02.2019).
15. Интерфейсы системы КАМАК [Электронный ресурс] // StudFiles. – URL: https://studfiles.net/preview/3305736/page:12/ (дата обращения 15.02.2019).
16. Комплекс приборов многомерной параметрической мессбауэровской спектроскопии. Философия проектирования [Электронный ресурс] // ЛАБОРАТОРИЯ МЕССБАУЭРОВСКОЙ СПЕКТРОСКОПИИ. – URL: http://iairas.ru/labs/res\_spectr\_dev.php/ (дата обращения 15.02.2019).
17. Training Custom Object Detector [Электронный ресурс] // TensorFlow Object Detection API tutorial. – URL: https://tensorflow-object-detection-api-tutorial.readthedocs.io/en/latest/training.html/ (дата обращения 15.02.2019).
18. ГОСТ 12.1.009-76 ССБТ. Система стандартов безопасности труда. Электробезопасность. Термины и определения. – М. : Изд-во стандартов, 1991. – 12 с.
19. Архив погоды в Екатеринбурге [Электронный ресурс] // Росгидрометцентр. – URL: [https://world-weather.ru/archive/­russia/yekaterinburg](https://world-weather.ru/archive/russia/yekaterinburg)/ (дата обращения 15.04.2019 г.).
20. СанПиН 2.2.2/2.4.1340-03. Гигиенические требования к персональным электронно-вычислительным машинам и организации работы. – М. : Изд-во стандартов, 1994. – 14 с.
21. СНиП 41-01-2003. Отопление, вентиляция и кондиционирование. – М. : Госстрой России, 2004. – 38 с.
22. ГОСТ 12.1.003-83. ССБТ. Шум. Общие требования безопасности. – М. : Стандартинформ, 2007. – 13 с.
23. СНиП 23-05-95. Естественное и искусственное освещение. – М. : Минстрой, 1995. – 10 с.
24. СанПиН 2.2.1/2.1.1.1278-03. Гигиенические требования к естественному, искусственному и совмещенному освещению жилых и общественных зданий. – М. : Изд-во стандартов, 1994. – 14 с.
25. ГОСТ 12.2.032-78. Система стандартов безопасности труда. Рабочее место при выполнении работ сидя. Общие эргономические требования. – М. : Изд-во стандартов, 1986. – 9 с.
26. НПБ 105-03. Определение категорий помещений, заданий и наружных установок по взрывопожарной и пожарной опасности. – М. : Стройиздат, 2003. – 16 с.
27. СНиП 21-01-97. Пожарная безопасность зданий и сооружений. Введ. 1998.01.01. – М.: ГУП ЦПП, 2002. – 25 с.
28. Барышев Е. Е., Тягунов Г. В., Фетисов И. Н. Безопасность жизнедеятельности: методические указания к оформлению раздела «Безопасность жизнедеятельности» в дипломных проектах и работах. – Екатеринбург : УрФУ, 2013. – 51 с.
29. Волкова А.А., Шишкунов Г.В., Тягунов Г.В. Безопасность жизнедеятельности: учебник. – Екатеринбург : УрФУ, 2013. – 55 с.
30. Арустамов Э.А., Левакова И.В., Баркалова Н.В. Экологические основы природопользования. М: «Дашков и К», 2008. – 320 с.
31. Российская Федерация. Законы. Об утверждении Правил обращения с отходами производства и потребления в части осветительных устройств, электрических ламп, ненадлежащие сбор, накопление, использование, обезвреживание, транспортирование и размещение которых может повлечь причинение вреда жизни, здоровью граждан, вреда животным, растениям и окружающей среде [Электронный ресурс] : федер. закон : [принят Гос. Думой 4 марта 1998 г.]. // Консультант Плюс. – URL: http://www.consultant.ru/document/cons\_doc\_LAW\_18254/ (дата обращения 15.05.2019).
32. Российская Федерация. Законы. Об энергосбережении и о повышении энергетической эффективности, и о внесении изменений в отдельные законодательные акты Российской Федерации [Электронный ресурс] : федер. закон : [принят Гос. Думой 23 ноября 2009 г.]. // Консультант Плюс. – URL: http://www.consultant.ru/document/cons\_doc\_LAW\_93978/ (дата обращения: 17.05.2019).

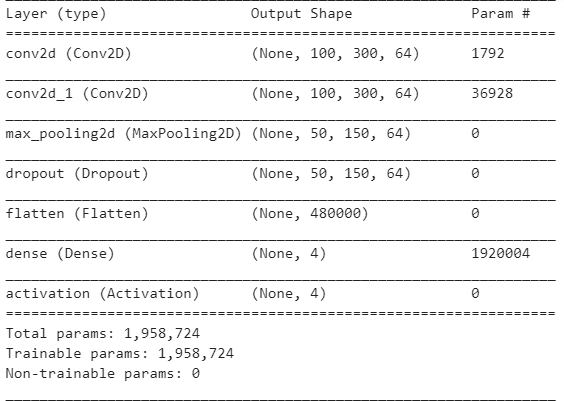
# ПРИЛОЖЕНИЕ А

Различные варианты структур НС [6]



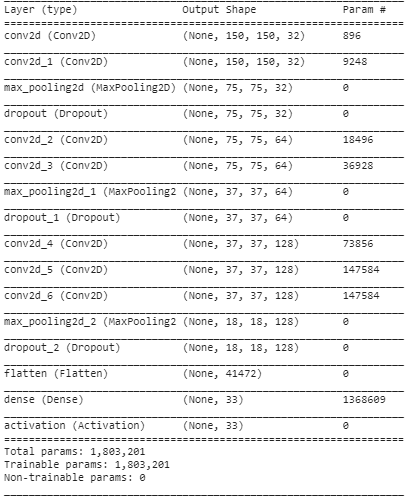
# ПРИЛОЖЕНИЕ Б

Структура нейронной сети для источника питания



# ПРИЛОЖЕНИЕ В

Структура нейронной сети для манометра



# ПРИЛОЖЕНИЕ Г

Листинг программы для конвертации файлов из формата .xml в .csv

*"""*

*Usage:*

*# Create train data:*

*python xml\_to\_csv.py -i [PATH\_TO\_IMAGES\_FOLDER]/train -o [PATH\_TO\_ANNOTATIONS\_FOLDER]/train\_labels.csv*

*# Create test data:*

*python xml\_to\_csv.py -i [PATH\_TO\_IMAGES\_FOLDER]/test -o [PATH\_TO\_ANNOTATIONS\_FOLDER]/test\_labels.csv*

*"""*

**import** **os**

**import** **glob**

**import** **pandas** **as** **pd**

**import** **argparse**

**import** **xml.etree.ElementTree** **as** **ET**

**def** xml\_to\_csv(path):

*"""Iterates through all .xml files (generated by labelImg) in a given directory and combines them in a single Pandas datagrame.*

*Parameters:*

*----------*

*path : {str}*

*The path containing the .xml files*

*Returns*

*-------*

*Pandas DataFrame*

*The produced dataframe*

*"""*

xml\_list = []

**for** xml\_file **in** glob.glob(path + '/\*.xml'):

tree = ET.parse(xml\_file)

root = tree.getroot()

**for** member **in** root.findall('object'):

value = (root.find('filename').text,

int(root.find('size')[0].text),

int(root.find('size')[1].text),

member[0].text,

int(member[4][0].text),

int(member[4][1].text),

int(member[4][2].text),

int(member[4][3].text)

)

xml\_list.append(value)

column\_name = ['filename', 'width', 'height',

'class', 'xmin', 'ymin', 'xmax', 'ymax']

xml\_df = pd.DataFrame(xml\_list, columns=column\_name)

**return** xml\_df

**def** main():

*# Initiate argument parser*

parser = argparse.ArgumentParser(

description="Sample TensorFlow XML-to-CSV converter")

parser.add\_argument("-i",

"--inputDir",

help="Path to the folder where the input .xml files are stored",

type=str)

parser.add\_argument("-o",

"--outputFile",

help="Name of output .csv file (including path)", type=str)

args = parser.parse\_args()

**if**(args.inputDir **is** None):

args.inputDir = os.getcwd()

**if**(args.outputFile **is** None):

args.outputFile = args.inputDir + "/labels.csv"

**assert**(os.path.isdir(args.inputDir))

xml\_df = xml\_to\_csv(args.inputDir)

xml\_df.to\_csv(

args.outputFile, index=None)

**print**('Successfully converted xml to csv.')

**if** \_\_name\_\_ == '\_\_main\_\_':

main()

*# Create train data:*

python xml\_to\_csv.py -i [PATH\_TO\_IMAGES\_FOLDER]/train -o [PATH\_TO\_ANNOTATIONS\_FOLDER]/train\_labels.csv

*# Create test data:*

python xml\_to\_csv.py -i [PATH\_TO\_IMAGES\_FOLDER]/test -o [PATH\_TO\_ANNOTATIONS\_FOLDER]/test\_labels.csv

*# For example*

*# python xml\_to\_csv.py -i C:\Users\sglvladi\Documents\TensorFlow\workspace\training\_demo\images\train -o C:\Users\sglvladi\Documents\TensorFlow\workspace\training\_demo\annotations\train\_labels.csv*

*# python xml\_to\_csv.py -i C:\Users\sglvladi\Documents\TensorFlow\workspace\training\_demo\images\test -o C:\Users\sglvladi\Documents\TensorFlow\workspace\training\_demo\annotations\test\_labels.csv*

# ПРИЛОЖЕНИЕ Д

Листинг программы для конвертации файлов из формата .csv в .record

*"""*

*Usage:*

*# Create train data:*

*python generate\_tfrecord.py --label=<LABEL> --csv\_input=<PATH\_TO\_ANNOTATIONS\_FOLDER>/train\_labels.csv --output\_path=<PATH\_TO\_ANNOTATIONS\_FOLDER>/train.record*

*# Create test data:*

*python generate\_tfrecord.py --label=<LABEL> --csv\_input=<PATH\_TO\_ANNOTATIONS\_FOLDER>/test\_labels.csv --output\_path=<PATH\_TO\_ANNOTATIONS\_FOLDER>/test.record*

*"""*

**from** **\_\_future\_\_** **import** division

**from** **\_\_future\_\_** **import** print\_function

**from** **\_\_future\_\_** **import** absolute\_import

**import** **os**

**import** **io**

**import** **pandas** **as** **pd**

**import** **tensorflow** **as** **tf**

**import** **sys**

sys.path.append("../../models/research")

**from** **PIL** **import** Image

**from** **object\_detection.utils** **import** dataset\_util

**from** **collections** **import** namedtuple, OrderedDict

flags = tf.app.flags

flags.DEFINE\_string('csv\_input', '', 'Path to the CSV input')

flags.DEFINE\_string('output\_path', '', 'Path to output TFRecord')

flags.DEFINE\_string('label', '', 'Name of class label')

*# if your image has more labels input them as*

*# flags.DEFINE\_string('label0', '', 'Name of class[0] label')*

*# flags.DEFINE\_string('label1', '', 'Name of class[1] label')*

*# and so on.*

flags.DEFINE\_string('img\_path', '', 'Path to images')

FLAGS = flags.FLAGS

*# TO-DO replace this with label map*

*# for multiple labels add more else if statements*

**def** class\_text\_to\_int(row\_label):

**if** row\_label == FLAGS.label: *# 'ship':*

**return** 1

*# comment upper if statement and uncomment these statements for multiple labelling*

*# if row\_label == FLAGS.label0:*

*# return 1*

*# elif row\_label == FLAGS.label1:*

*# return 0*

**else**:

None

**def** split(df, group):

data = namedtuple('data', ['filename', 'object'])

gb = df.groupby(group)

**return** [data(filename, gb.get\_group(x)) **for** filename, x **in** zip(gb.groups.keys(), gb.groups)]

**def** create\_tf\_example(group, path):

**with** tf.gfile.GFile(os.path.join(path, '{}'.format(group.filename)), 'rb') **as** fid:

encoded\_jpg = fid.read()

encoded\_jpg\_io = io.BytesIO(encoded\_jpg)

image = Image.open(encoded\_jpg\_io)

width, height = image.size

filename = group.filename.encode('utf8')

image\_format = b'jpg'

*# check if the image format is matching with your images.*

xmins = []

xmaxs = []

ymins = []

ymaxs = []

classes\_text = []

classes = []

**for** index, row **in** group.object.iterrows():

xmins.append(row['xmin'] / width)

xmaxs.append(row['xmax'] / width)

ymins.append(row['ymin'] / height)

ymaxs.append(row['ymax'] / height)

classes\_text.append(row['class'].encode('utf8'))

classes.append(class\_text\_to\_int(row['class']))

tf\_example = tf.train.Example(features=tf.train.Features(feature={

'image/height': dataset\_util.int64\_feature(height),

'image/width': dataset\_util.int64\_feature(width),

'image/filename': dataset\_util.bytes\_feature(filename),

'image/source\_id': dataset\_util.bytes\_feature(filename),

'image/encoded': dataset\_util.bytes\_feature(encoded\_jpg),

'image/format': dataset\_util.bytes\_feature(image\_format),

'image/object/bbox/xmin': dataset\_util.float\_list\_feature(xmins),

'image/object/bbox/xmax': dataset\_util.float\_list\_feature(xmaxs),

'image/object/bbox/ymin': dataset\_util.float\_list\_feature(ymins),

'image/object/bbox/ymax': dataset\_util.float\_list\_feature(ymaxs),

'image/object/class/text': dataset\_util.bytes\_list\_feature(classes\_text),

'image/object/class/label': dataset\_util.int64\_list\_feature(classes),

}))

**return** tf\_example

**def** main(\_):

writer = tf.python\_io.TFRecordWriter(FLAGS.output\_path)

path = os.path.join(os.getcwd(), FLAGS.img\_path)

examples = pd.read\_csv(FLAGS.csv\_input)

grouped = split(examples, 'filename')

**for** group **in** grouped:

tf\_example = create\_tf\_example(group, path)

writer.write(tf\_example.SerializeToString())

writer.close()

output\_path = os.path.join(os.getcwd(), FLAGS.output\_path)

**print**('Successfully created the TFRecords: {}'.format(output\_path))

**if** \_\_name\_\_ == '\_\_main\_\_':

tf.app.run()

*# Create train data:*

python generate\_tfrecord.py --label=<LABEL> --csv\_input=<PATH\_TO\_ANNOTATIONS\_FOLDER>/train\_labels.csv

--img\_path=<PATH\_TO\_IMAGES\_FOLDER>/train --output\_path=<PATH\_TO\_ANNOTATIONS\_FOLDER>/train.record

*# Create test data:*

python generate\_tfrecord.py --label=<LABEL> --csv\_input=<PATH\_TO\_ANNOTATIONS\_FOLDER>/test\_labels.csv

--img\_path=<PATH\_TO\_IMAGES\_FOLDER>/test

--output\_path=<PATH\_TO\_ANNOTATIONS\_FOLDER>/test.record

*# For example*

*# python generate\_tfrecord.py --label=ship --csv\_input=C:\Users\sglvladi\Documents\TensorFlow\workspace\training\_demo\annotations\train\_labels.csv --output\_path=C:\Users\sglvladi\Documents\TensorFlow\workspace\training\_demo\annotations\train.record --img\_path=C:\Users\sglvladi\Documents\TensorFlow\workspace\training\_demo\images\train*

*# python generate\_tfrecord.py --label=ship --csv\_input=C:\Users\sglvladi\Documents\TensorFlow\workspace\training\_demo\annotations\test\_labels.csv --output\_path=C:\Users\sglvladi\Documents\TensorFlow\workspace\training\_demo\annotations\test.record --img\_path=C:\Users\sglvladi\Documents\TensorFlow\workspace\training\_demo\images\test*

# ПРИЛОЖЕНИЕ Е

Листинг программы для обучения НС

model {

ssd {

num\_classes: 1#Количество объектов для распознавания

box\_coder {

faster\_rcnn\_box\_coder {

y\_scale: 10.0

x\_scale: 10.0

height\_scale: 5.0

width\_scale: 5.0

}

}

matcher {

argmax\_matcher {

matched\_threshold: 0.5

unmatched\_threshold: 0.5

ignore\_thresholds: false

negatives\_lower\_than\_unmatched: true

force\_match\_for\_each\_row: true

}

}

similarity\_calculator {

iou\_similarity {

}

}

anchor\_generator {

ssd\_anchor\_generator {

num\_layers: 6

min\_scale: 0.2

max\_scale: 0.95

aspect\_ratios: 1.0

aspect\_ratios: 2.0

aspect\_ratios: 0.5

aspect\_ratios: 3.0

aspect\_ratios: 0.3333

reduce\_boxes\_in\_lowest\_layer: true

}

}

image\_resizer {

fixed\_shape\_resizer {

height: 300

width: 300

}

}

box\_predictor {

convolutional\_box\_predictor {

min\_depth: 0

max\_depth: 0

num\_layers\_before\_predictor: 0

use\_dropout: false

dropout\_keep\_probability: 0.8

kernel\_size: 3

box\_code\_size: 4

apply\_sigmoid\_to\_scores: false

conv\_hyperparams {

activation: RELU\_6,

regularizer {

l2\_regularizer {

weight: 0.00004

}

}

initializer {

truncated\_normal\_initializer {

stddev: 0.03

mean: 0.0

}

}

}

}

}

feature\_extractor {

type: ‘ssd\_resnet\_50\_fpn\_coco’ # название выбранной НС

min\_depth: 16

depth\_multiplier: 1.0

conv\_hyperparams {

activation: RELU\_6,

regularizer {

l2\_regularizer {

weight: 0.00004

}

}

initializer {

truncated\_normal\_initializer {

stddev: 0.03

mean: 0.0

}

}

batch\_norm {

train: true,

scale: true,

center: true,

decay: 0.9997,

epsilon: 0.001,

}

}

override\_base\_feature\_extractor\_hyperparams: true

}

loss {

classification\_loss {

weighted\_sigmoid {

}

}

localization\_loss {

weighted\_smooth\_l1 {

}

}

hard\_example\_miner {

num\_hard\_examples: 3000

iou\_threshold: 0.99

loss\_type: CLASSIFICATION

max\_negatives\_per\_positive: 3

min\_negatives\_per\_image: 0

}

classification\_weight: 1.0

localization\_weight: 1.0

}

normalize\_loss\_by\_num\_matches: true

post\_processing {

batch\_non\_max\_suppression {

score\_threshold: 1e-8

iou\_threshold: 0.6

max\_detections\_per\_class: 100

max\_total\_detections: 100

}

score\_converter: SIGMOID

}

}

}

train\_config: {

batch\_size: 12 # этот параметр влияет на скорость обучния

optimizer {

rms\_prop\_optimizer: {

learning\_rate: {

exponential\_decay\_learning\_rate {

initial\_learning\_rate: 0.004

decay\_steps: 800720

decay\_factor: 0.95

}

}

momentum\_optimizer\_value: 0.9

decay: 0.9

epsilon: 1.0

}

}

fine\_tune\_checkpoint: "pre-trained-model/model.ckpt" # путь к файлам предобученной НС

from\_detection\_checkpoint: true

num\_steps: 200000

data\_augmentation\_options {

random\_horizontal\_flip {

}

}

data\_augmentation\_options {

ssd\_random\_crop {

}

}

}

train\_input\_reader: {

tf\_record\_input\_reader {

input\_path: "annotations/train.record" # путь к тренировочному TFRecord файлу

}

label\_map\_path: "annotations/label\_map.pbtxt" # путь к карте меток

}

eval\_config: {

num\_examples: 8000

max\_evals: 10

}

eval\_input\_reader: {

tf\_record\_input\_reader {

input\_path: "annotations/test.record"

}

label\_map\_path: "annotations/label\_map.pbtxt"

shuffle: false

num\_readers: 1

}