

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ рОССИЙСКОЙ ФЕДЕРАЦИИ

Федеральное государственное автономное образовательное учреждение высшего образования

**«Дальневосточный федеральный университет»**

|  |  |
| --- | --- |
| **ШКОЛА ЕСТЕСТВЕННЫХ НАУК**  **кафедра информатики, математического и компьютерного моделирования** | |
| «Доработка библиотеки нейронных сетей» |

**КУРСОВАЯ РАБОТА**

по образовательной программе подготовки бакалавров

по направлению 01.03.02 «Прикладная математика и информатика»

|  |  |
| --- | --- |
| Работа защищена  с оценкой \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  Регистрационный номер \_\_\_\_\_\_\_  «\_\_\_\_»\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_20\_\_\_г. | Студентка группы № Б8117-01.03.02  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Глушкова Д.И.  (подпись)  «\_\_\_\_\_» \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ 20\_\_\_\_г.  Руководитель \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  (должность, ученое звание)  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  (подпись) (ФИО)  «\_\_\_\_\_\_»\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_20\_\_\_г. |

г. Владивосток

2020

**Оглавление**

[Аннотация 3](#_Toc44095006)

[1. Введение 4](#_Toc44095007)

[1.1 Глоссарий 4](#_Toc44095008)

[1.2 Описание предметной области 4](#_Toc44095009)

[1.3 Неформальная постановка задачи 10](#_Toc44095010)

[1.4 Обзор существующих решений 10](#_Toc44095011)

[1.5 План работ 11](#_Toc44095012)

[2. Требование к окружению 12](#_Toc44095013)

[2.1 Требования к аппаратному обеспечению 12](#_Toc44095014)

[2.2 Требования к программному обеспечению 12](#_Toc44095015)

[2.3 Требования к пользователям 12](#_Toc44095016)

[3. Спецификация данных 13](#_Toc44095017)

[4. Функциональные требования 14](#_Toc44095018)

[4.1 Задача 1. Система должна 14](#_Toc44095019)

[4.2 Задача 2. Система должна 14](#_Toc44095020)

[4.3 Библиотека подпрограмм (классов). Требования к интерфейсу 14](#_Toc44095021)

[5. Проект 15](#_Toc44095022)

[5.1 Средства реализации 15](#_Toc44095023)

[5.2 Структуры данных 15](#_Toc44095024)

[5.3 Модули и алгоритмы 15](#_Toc44095025)

[5.4 . Стандарт кодирования 15](#_Toc44095026)

[5.5 Проект интерфейса 15](#_Toc44095027)

[6. Реализация и тестирование 16](#_Toc44095028)

[Заключение. 17](#_Toc44095029)

[Список литературы. 18](#_Toc44095030)

# Аннотация

Целью данной работы является разработка веб-интерфейса для фреймворка нейронных сетей с функционалом для подготовки данных для обучения и расширения датасета, визуализации процесса обучения нейросети и с возможностью контроля версий результатов обучения в зависимости от входных данных.

# Введение

## **Глоссарий**

1. average precisions –
2. CUDA
3. Вес -
4. Глубокое обучение – мощный набор техник обучения нейросетей.
5. Датасет -
6. Нейрон – это вычислительная единица, которая получает информацию, производит над ней простые вычисления и передает ее дальше. нейроны оперируют числами в диапазоне [0,1] или [-1,1].
7. Нейронная сеть — это последовательность нейронов, соединенных между собой синапсами.
8. Операторы управления потоком – повышают производительность и универсальность сивольной системы
9. Персептрон – это система, имитирующая все процессы внутри головного мозга.
10. Синапс – это связь между двумя нейронами. У синапсов есть 1 параметр —
11. вес.
12. Функция потерь -
13. Эпоха –

## Описание предметной области

Теория распознавания образов включает в себя разработку методов идентификации предметов, явлений и сигналов. Потребность в таком распознавании возникает во многих областях, на нем основано машинное зрение, которое активно внедряется на производство. Однако его возможности ограничиваются качеством и точностью распознавания.

Проблема распознавания сотрудника исключительно данной компании заключается в детектировании объекта на изображении из множества категорий, например, любой человек может являться как сотрудником этой компании, так и соседнего магазина. Сложности задачи заключается в том, что распределение данных по категориям чаще всего неравномерное и в каких-то случаях различия между классами могут быть мелкие.

На помощь для решения таких узконаправленных задач, как описано выше, приходят нейронные сети. Нейронные сети являются одним из направлений научных исследований в области создания искусственного интеллекта, в основе которого лежит стремление подражать нервной системе человека.

Глубокое обучение – это подмножество нейронных сетей (рис.1). Оно позволяет обучать модель и предсказывать результат по набору входных данных. Определяющая особенность глубокого обучения – наличие нескольких слоев, разделяющих вход и выход (рис. 2).

В рамках проекта по распознаванию монет для сбербанка от ШЦЭ (2019 г.) в качестве фреймворка был использован darknet. Darknet — это активно развивающийся бесплатный фреймворк с открытым исходным кодом, написанный на языке C. Его поддержка производится регулярно: на данный

Рисунок Схема мира

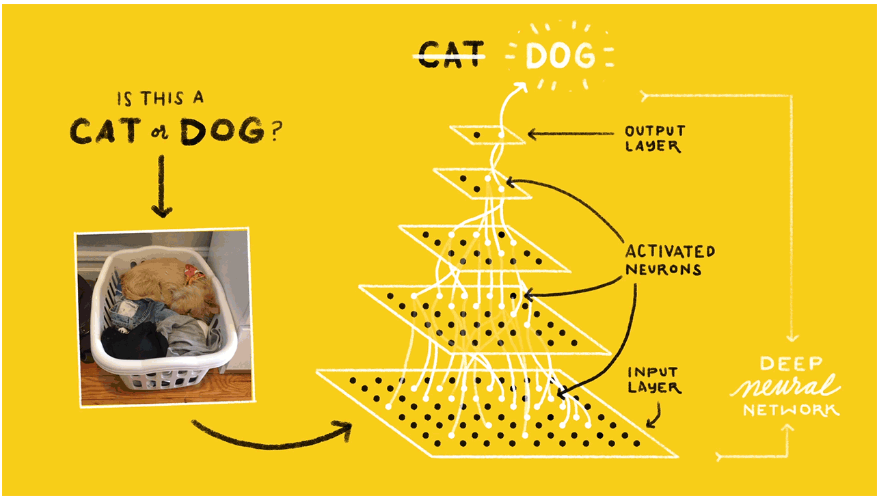
момент 1892 коммита на ветке мастера, более одиннадцати тысяч звезд на репозитории и более тринадцати тысяч форков. Он быстрый и лёгкий в использовании. Но в работе с ним возникал ряд неудобств, из которых можно выделить:

Рисунок Визуализация работы глубокой нейросети

* На графике, отображающем результаты обучения, нет названия обучающегося проекта. Из этого факта выделяется проблема:
  + При одновременном запуске нескольких процессов обучения, не представляется возможным соотнести график с соответствующим процессом (см. рис.№5. График результатов обучения. Даркнет)
* При прерывании процесса обучения, график, который был до остановки процесса, не сохраняется и отображается с текущей точки запуска;
* ???Обучали нейронную сеть датасетом, там пишется все в файл chart.png и он один, там нет chart01.png, когда два процесса обучения, они пишут в один файл и если обучаешь, а потом запускаешь новое обучение с расширенным датасетом, старый chart.png не сохраняется, а перезаписывается на новый график. А мы хотим сравнивать эти графики.???

Исследованы следующие этапы работ с нейронными сетями:

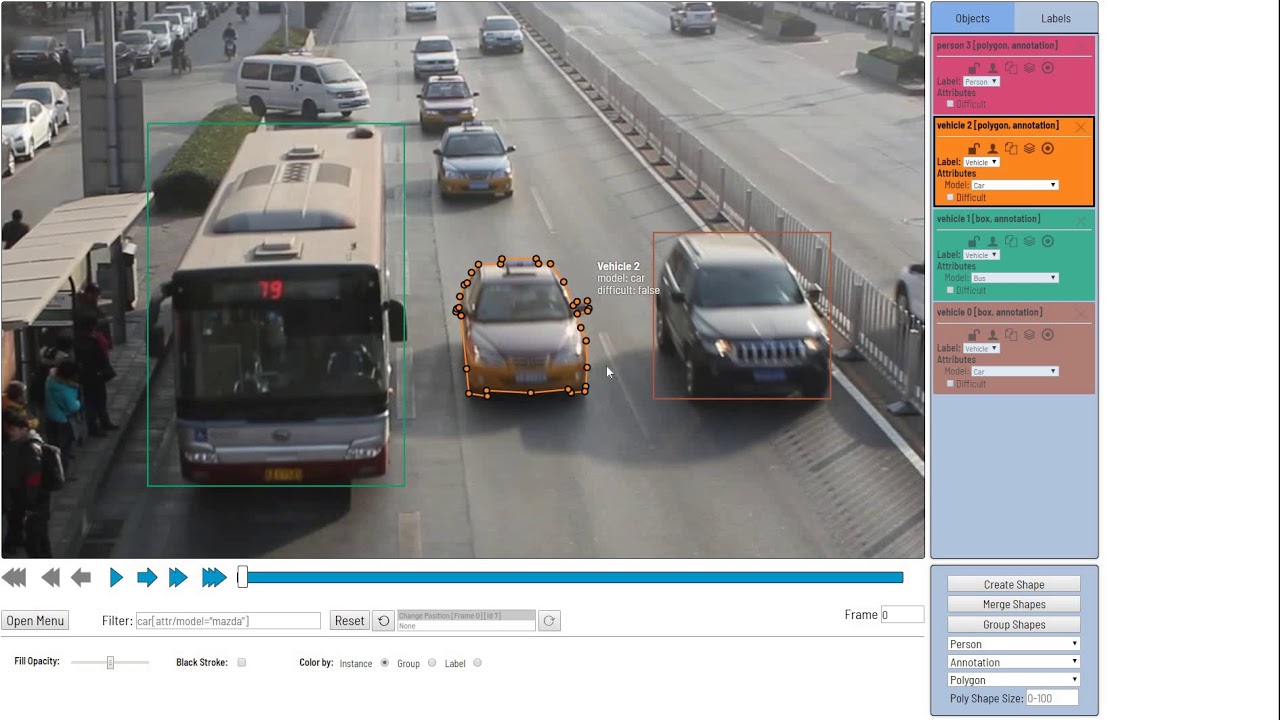
* Первый этап – сбор данных. Для обучения требуется большой датасет, который можно собрать самостоятельно, а можно взять готовые. Сделать это можно используя, например, Google Dataset, который по ключевому слову позволяет искать датасеты по всей сети.
  + Составление собственного датасета. Выставляются следующие требования к набору данных:

Рисунок Разметка данных с помощью Cvat

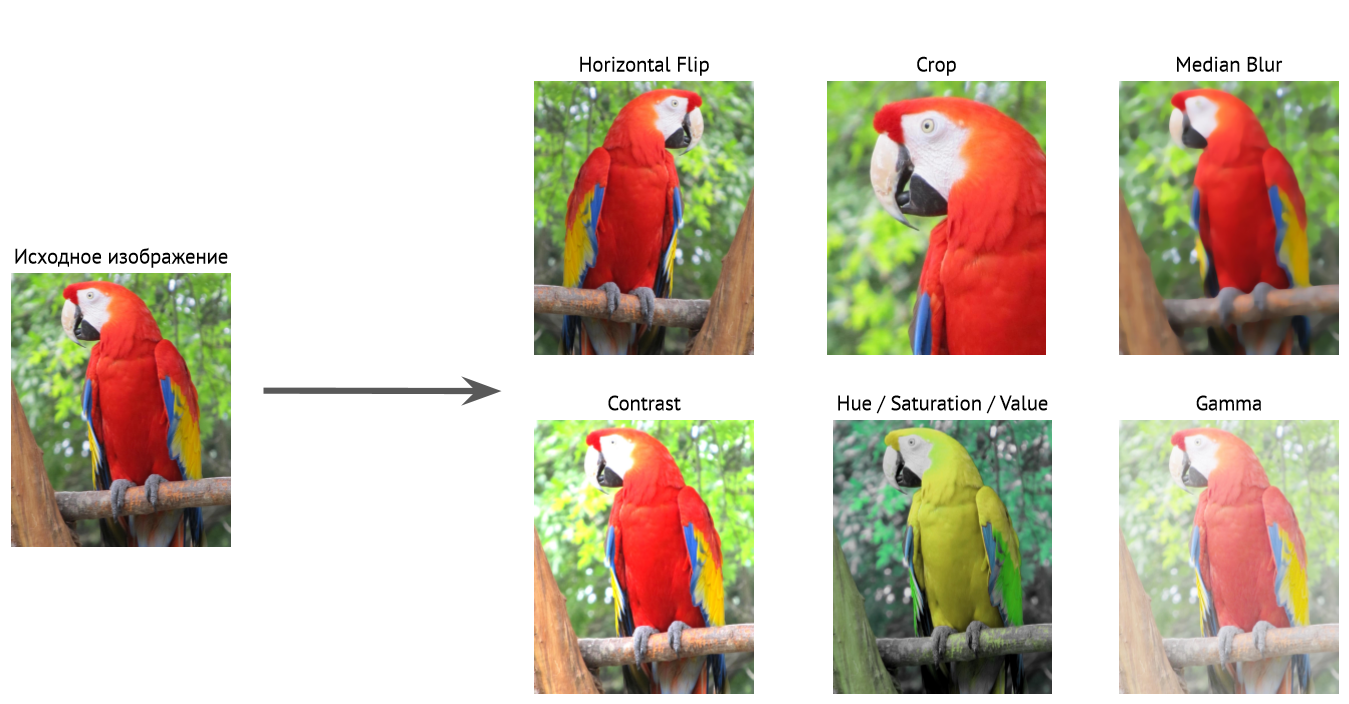
* + 1. Количество. Объем датасета составляет несколько тысяч изображений.
  + 2. Разметка данных. Производим разметку объектов, назначая им соответствующие категории.
  + 3. Балансировка. При составлении датасета важно не допустить переобучения. Чтобы избежать этого, нужно держать баланс между количеством различных данных по каждой категории.
* Лучший способ получения датасета, согласно вышеперечисленным критериям, – отснять видео объекта в различных условиях на различных фонах, распознаванию которого должна обучиться нейросеть. Затем, в соответствии с вышеописанным пунктом №2, приступить к разметке кадров из видео, взятых, условно, с шагом h = 5 кадрам. Мною использовалась программа Computer Vision Annotation Tool (cvat).
* Второй этап – аугментация (рис.4). Под аугментацией подразумевается генерирование новых данных на основе имеющихся, что позволяет довольно просто и дешево расширить датасет при помощи «подручных» средств: добавления размытия, зернистости, и т.д. . Производить аугментацию можно с помощью следующих библиотек:

Рисунок Аугментация

* + Pillow. Это достаточно популярная библиотека для языка Python. Это легкая в использовании библиотека, которая умеет многое, например:
    - Накладывать одно изображение поверх другого
    - Поворот изображений
    - Применение множества различных фильтров
    - Изменять размер
  + Scikit-image. Известная Python-библиотека для обработки изображений, работающая с массивами numpy, представляющими собой набор алгоритмов обработки изображений. Имеет такие алгоритмы как, например:
    - Сегментации
    - Геометрических преобразований
    - Фильтрации
    - Морфологии
    - Обнаружения признаков
* Третий этап – обучение. Использовалось глубокое обучение, описание которого было приведено выше.
  + Обучение нейронной сети выполняется аналогично машинному обучению – берется датасет, сравниваются входные данные с желаемыми выходными, формируется вектор ошибок и на их основе применяются правки к сети.

В процессе обучения можно наблюдать график по показателям обучения, функций ошибок:

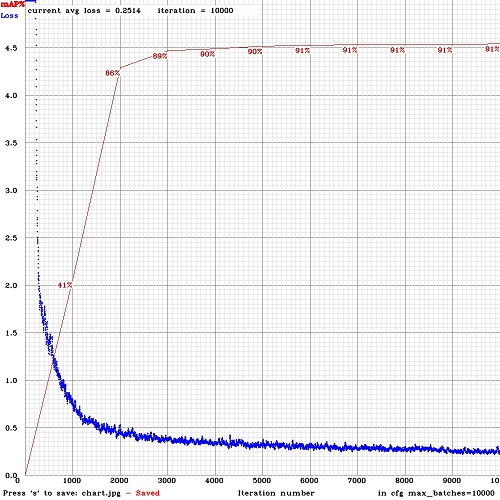
* + Красной линией отмечено среднее значение average precisions. Обучение продолжается, пока эта величина увеличивается.
  + Синей линией отмечена функция потерь, обучение нужно продолжать, пока ее значение != 0.24
  + Изображенный на графике процесс обучения прошел 10000 итераций (эпох).

Рисунок График результатов обучения. Даркнет

## Неформальная постановка задачи

Система должна предоставлять пользователю следующие возможности:

* Подготовка данных для обучения
* Аугментация данных
* Балансировка данных
* Запуск процесса обучения на сервере
* Визуализация процесса обучения в realtime
* Контроль версий результатов обучения в зависимости от входных данных

## Обзор существующих решений

Программы для глубокого обучения пишутся как с нуля, так и с использованием фреймворков, что гораздо эффективнее. Проведем сравнение между известными фреймворками и результатом данной курсовой работы.

|  |  |  |  |
| --- | --- | --- | --- |
| Характеристика | Theano | TensorFlow | Курсовая работа |
| Открытый исходный код | + | + | + |
| Платформа | Кросс-платформенное решение | Linux, Mac OS X, планируется поддержка Windows | Windows, Linux, Mac OS |
| Язык программирования | Python | C++  Python | С  Python |
| Поддержка CUDA | + | + | + |
| Наличие предварительно обученных моделей | + | - | + |
| Производительность | Отлично | Средне |  |
| Память | Отлично | Средне |  |

**??? тут наверняка должна быть какая нибудь строка ???**

## План работ

1. Изучение предметной области
2. Освоение спецификации разметки данных
3. Исследование методов для расширения датасета. Аугментация.
4. Обучение нейросети на составленном датасете.
5. Исправление части выявленных недостатков фреймворка (всм пауза и перемотка)
6. Разработка веб-интерфейса для обучения нейронных сетей
   1. Разработка серверной части
      1. Запуск тестирования
      2. Версионирование
   2. Разработка клиентской части

# Требование к окружению

## Требования к аппаратному обеспечению

* Процессор с тактовой частотой не менее 3,5 ГГц
* Видеокарта не слабее Nvidia GTX960
* Оперативная память не менее 16 гб

## Требования к программному обеспечению

* Windows or Linux or Mac OS
* Python 3.7
* Django 3.0.7
* darknet\_0.3
* opencv\_python-4.2.0.34
* scikit\_image-0.17.2
* CMake >= 3.12: https://cmake.org/download/
* CUDA 10.0
* OpenCV >= 2.4
* cuDNN >= 7.0 for CUDA 10.0
* GPU with CC >= 3.0:
* Браузер, оснащенный поддержкой JS, например, Firefox 77.0.1

## **Требования к пользователям**

Результат выполнения курсовой работы предназначен для использования программистами в целях обучения нейронных сетей.

# Спецификация данных

# Функциональные требования

## Задача 1. Система должна

* Предоставлять возможность остановки видео воспроизведения
* Позволять пользователю перематывать вперед/назад видео поток

## Задача 2. Система должна

* Предоставлять возможность для:
* Составления/загрузки данных
* Аугментации данных
* Балансировк данных
* Уметь запускать процесс обучения на сервере
* Уметь визуализировать процесс обучения в режиме реального времени
* Иметь возможность запуска без обучения
* Предоставлять возможность для версионирования

## Библиотека подпрограмм (классов). Требования к интерфейсу

# Проект

## Средства реализации

Для реализации проекта были выбраны библиотеки darknet, opencv, scikit\_image и фреймворк Django. Darknet – это связка для библиотеки yolo (v4), которая, в свою очередь, представляет собой систему обнаружения объектов в реальном времени. Opencv – библиотека алгоритмов компьютерного зрения, обработки изображений и численных алгоритмов общего назначения с открытым кодом, а scikit\_image – это пакет Python для обработки изображений, который работает с массивами numpy, представляющими собой набор алгоритмов обработки изображений. Все перечисленные библиотеки являются удобными и легкими в использовании обертками, предназначенными для работы с изображениями. Выбор языка программирования Python обусловлен наличием этих основных библиотек, а также легкостью работы с фреймворком Django.

## Структуры данных

## Модули и алгоритмы

## . Стандарт кодирования

## Проект интерфейса

# Реализация и тестирование

В ходе работы было написано около 500 строк кода, примерно 100 на языке С, остальные – Python. Было произведено тестирование системы и реализован прототип.

Тестирование проводилось в несколько этапов:

* Модульное тестирование. Использовались датасеты, предоставленные <https://github.com/AlexeyAB/darknet> для обучение нейросети через созданный веб-интерфейс
* Интеграционное тестирование. Эмитировались различные сценарии работ с приложением приглашенными пользователями.

# Заключение.

В процессе выполнения курсовой работы мною была освоена спецификация разметки данных, изучены методы расширения датасетов. Освоено глубокое обучение нейронных сетей.

Мною был повышен навык владения языками программирования Python & C, получен опыт написания клиентской и серверной части с помощью фреймворка Django.

Созданная мною система исправила часть выявленных недостатков фреймворка Darknet, а также добавила возможность работы через веб-сервис.

Дальнейшая разработка включает в себя интеграцию новых возможностей для моего веб-сервиса, что сделает его более функциональным в использовании. А также максимальное упрощение интерфейса, чтобы им мог воспользоваться не только программист, но и человек с базовыми знаниями программирования.

Большинство задач, поставленных в ходе моей работы, выполнены успешно.

# Список литературы.

1. <https://habr.com/ru/company/microsoft/blog/313318/>
2. <https://datasetsearch.research.google.com/>
3. <https://vas3k.ru/blog/machine_learning/>
4. <https://github.com/AlexeyAB/darknet>