

ПРАВИТЕЛЬСТВО РОССИЙСКОЙ ФЕДЕРАЦИИ
НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ
«ВЫСШАЯ ШКОЛА ЭКОНОМИКИ»

Факультет компьютерных наук
Департамент программной инженерии

СОГЛАСОВАНО

Доцент департамента программной
инженерии факультета компьютерных
наук

_____ Р.А. Родригес Залепинос
« ____ » _____ 2018 г.

УТВЕРЖДАЮ

Академический руководитель
образовательной программы
«Программная инженерия», канд. техн.
наук, профессор ДПИ ФКН

_____ В.В. Шилов
« ____ » _____ 2018 г.

Программа обработки растровых данных с помощью TensorFlow

Пояснительная записка

ЛИСТ УТВЕРЖДЕНИЯ

RU.17701729.04.16-01 81 01-1-ЛУ

Исполнитель

Студент группы БПИ177

_____ / Д. А. Потапенков /
« ____ » _____ 2018 г.

Москва 2018

Инв. № подл		Подп. и дата	
Взам. инв. №		Инв. № дубл.	
Подп. и дата		Подп. и дата	

УТВЕРЖДЕН

RU.17701729.04.16-01 81 01-1-ЛУ

Программа обработки растровых данных с помощью TensorFlow

Пояснительная записка

RU.17701729.04.16-01 81 01-1

Листов 12

Инв. № подл	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата

Москва 2018

ОГЛАВЛЕНИЕ

1. ВВЕДЕНИЕ	3
1.1. Наименование программы	3
1.2. Основание для разработки	3
2. НАЗНАЧЕНИЕ И ОБЛАСТЬ ПРИМЕНЕНИЯ	4
2.1. Функциональное назначение	4
2.2. Краткая характеристика области применения программы.....	4
3. ТЕХНИЧЕСКИЕ ХАРАКТЕРИСТИКИ	5
3.1. Постановка задачи на разработку программы	5
3.2. Описание алгоритма и функционирования программы.....	5
3.3. Описание и обоснование выбора метода организации входных и выходных данных	8
3.4. Описание и обоснование выбора состава технических и программных средств.	8
4. ТЕХНИКО-ЭКОНОМИЧЕСКИЕ ПОКАЗАТЕЛИ	9
4.1. Предполагаемая потребность	9
4.2. Ориентировочная экономическая эффективность	9
4.3. Экономические преимущества разработки по сравнению с отечественными и	
зарубежными аналогами	9
5. ПРИЛОЖЕНИЕ.....	10
5.1. Источники, использованные при разработке	Ошибка! Закладка не определена.
5.2. Описание и функциональное назначение классов и библиотек	10
5.3. Описание и функциональное назначение методов, полей и свойств	10
ЛИСТ РЕГИСТРАЦИИ ИЗМЕНЕНИЙ	12

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729. 04.16-01 81 01-1				
Инв. № подл.	Подп. и дата	Взам. Инв. №	Инв. № дубл.	Подп. и дата

1. ВВЕДЕНИЕ

1.1. Наименование программы

Наименование программы: «Программа обработки растровых данных с помощью TensorFlow» («Program for Raster Data Processing Using TensorFlow»).

1.2. Основание для разработки

Основанием на разработку является приказ декана факультета компьютерных наук Национального исследовательского университета «Высшая школа экономики» № 2.3-02/1212-01 от 12.12.2017 "Об утверждении тем, руководителей курсовых работ студентов образовательной программы Программная инженерия факультета компьютерных наук".

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729. 04.16-01 81 01-1				
Инв. № подл.	Подп. и дата	Взам. Инв. №	Инв. № дубл.	Подп. и дата

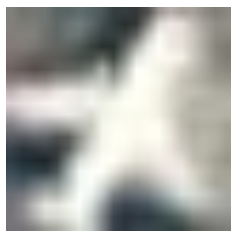
2. НАЗНАЧЕНИЕ И ОБЛАСТЬ ПРИМЕНЕНИЯ

2.1. Функциональное назначение

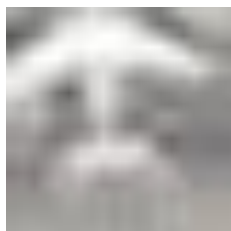
Программа представляет из себя клиент-серверное приложение, позволяющее пользователю отправить растровые спутниковые снимки, и получать обратно результаты распознавания снимка.

2.2. Краткая характеристика области применения программы

Приложение предназначено для отправки спутниковых снимков на сервер, где обученная модель распознает, изображен на снимке целый самолет (рис 1) или нет (рис 2), и отправляет обратно клиенту результат. Спутниковые снимки представляют из себя файлы в формате (.png, RGB, 20x20), можно загружать снимки других форматов, но результат распознавания может быть неверным.



(Рис 1) Самолёт



(Рис 2) Не самолёт

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729. 04.16-01 81 01-1				
Инв. № подл.	Подп. и дата	Взам. Инв. №	Инв. № дубл.	Подп. и дата

3. ТЕХНИЧЕСКИЕ ХАРАКТЕРИСТИКИ

3.1. Постановка задачи на разработку программы

Нужно реализовать клиент-серверное приложение, где серверная часть, написанная с использованием Tensorflow, распознает объекты на спутниковых снимках.

3.2. Описание алгоритма и функционирования программы

Работа программы состоит из нескольких блоков:

- Блок строения и обучения нейронной сети;
- Блок работы клиентского приложения;
- Блок работы сервера;

3.2.1. Блок обучения нейронной сети

Распознавание изображения происходит с помощью сверточной нейронной сети. Для построения нейронной сети были изучены: функции пакета TensorFlow, различные статьи по использованию нейросетей для распознавания изображений. В итоге было решено построить такую нейросеть :

1. Свёрточный слой#1: Применяет 32 фильтра 5x5, с функцией активации ReLU
2. Слой подвыборки#1: Объединяет фрагменты 2x2 в одно значение выбирая максимальный
3. Свёрточный слой#2: Применяет 64 фильтра 5x5, с функцией активации ReLU
4. Слой подвыборки#2: Опять объединяет фрагменты 2x2 в одно значение выбирая максимальный
5. Плотный слой#1: состоящий из 1024 нейронов
6. Плотный слой#2: Состоящий из 2 нейронов (Самолет, не самолет)

Тренировка сети была проведена с количеством шагов равным 1000.

Данные для обучения были взяты с сайта Kegggle [1], где они были даны в 2 форматах (данные одинаковые):

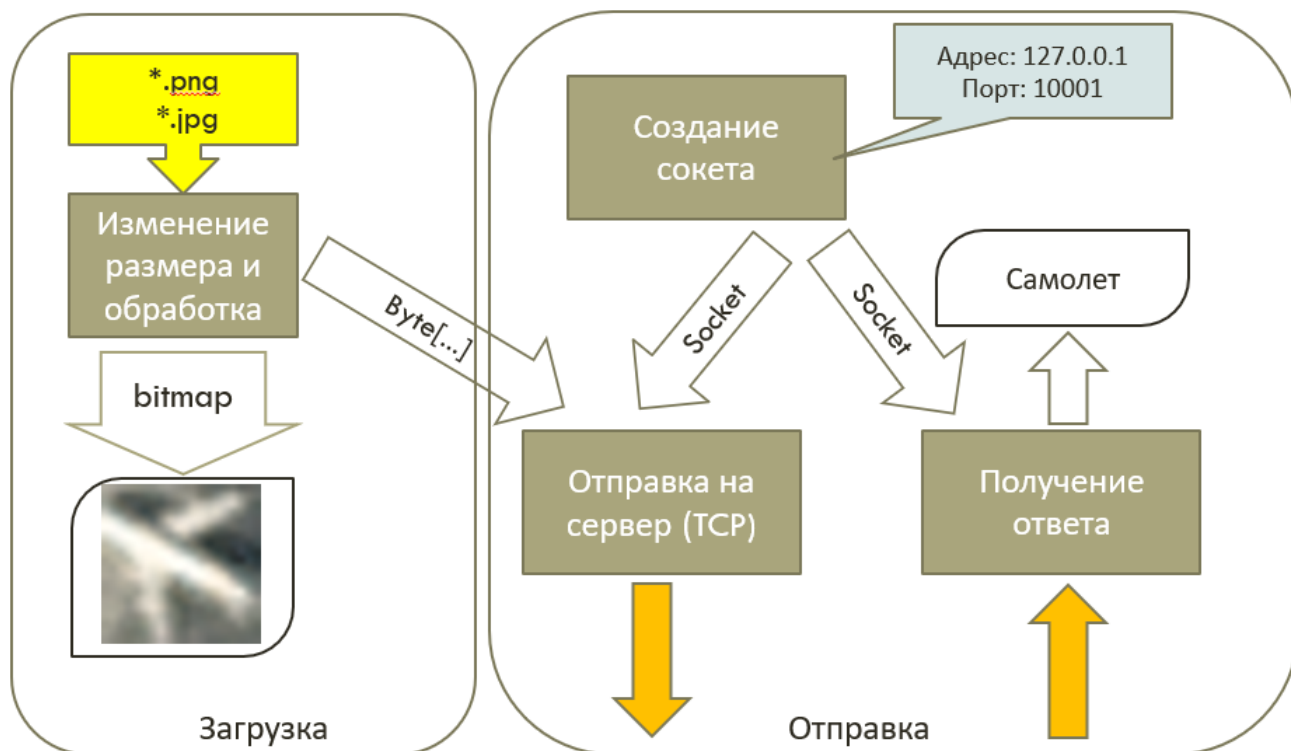
1. 32000 фотографий формата *.png размером 20 на 20 пикселей. Название фотографии в виде «Класс_IDсцены_координаты».
2. Текстовый файл в формате JSON. В нем содержатся метки “data” и “labels”. В “data” хранится массив значения пикселей для каждой RGB картинке 20x20 в массивах состоящих из 1200 целых значений (400 красных, затем 400 зелёных, и 400 синих). Изображения хранятся в строчном порядке, так что первые 20 значений являются красными значениями первой строки.

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729. 04.16-01 81 01-1				
Инв. № подл.	Подп. и дата	Взам. Инв. №	Инв. № дубл.	Подп. и дата

Для тренировки были выбраны данные из формата JSON из-за простоты работы с ними. Снимки из формата RGB были переведены в черно-белый и обработаны фильтром контрастности для лучшего обучения.

Для обучения были взяты 7000 изображений самолетов и 21000 не самолетов. Для проверки были взяты оставшиеся 1000 самолетов и 300 не самолетов

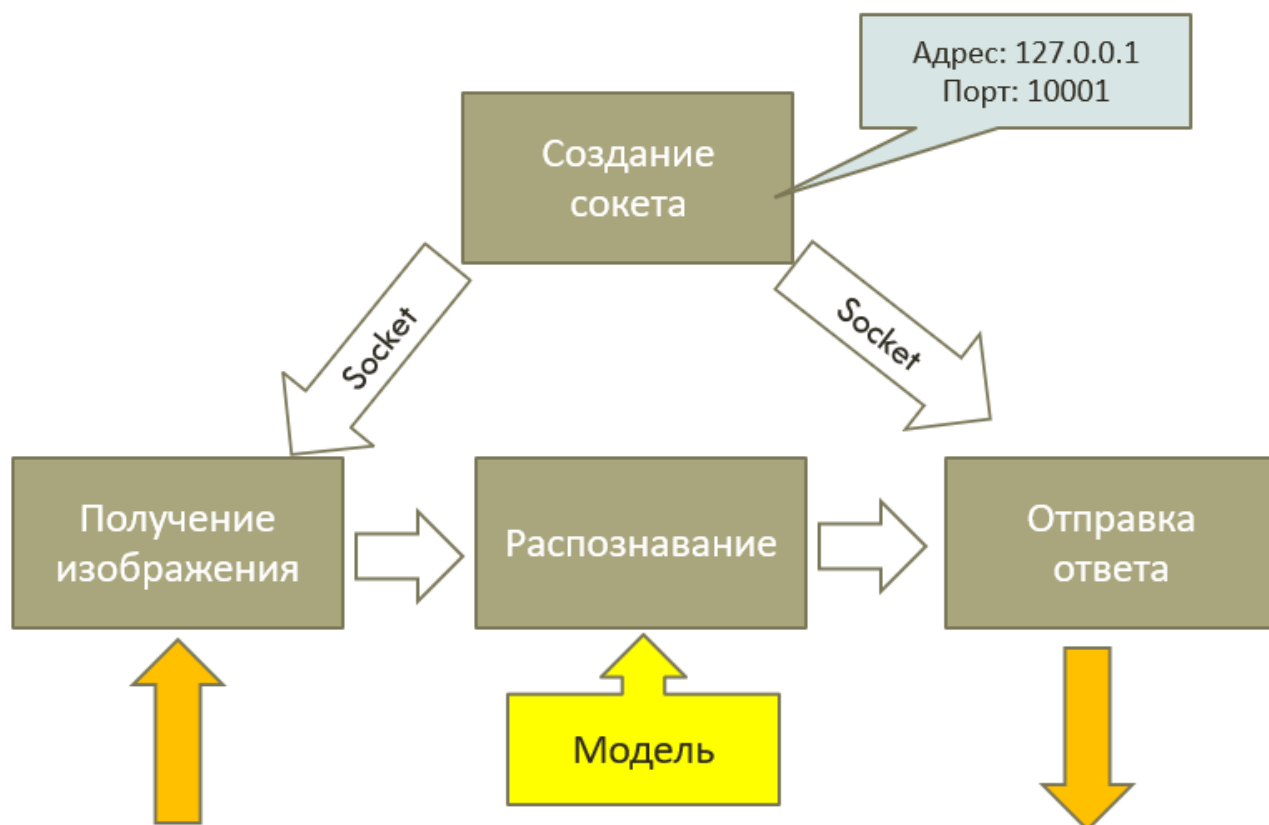
3.2.2. Блок работы клиентского приложения



1. При нажатии на кнопку «Загрузить», появляется окно выбора картинки.
2. Выбранная картинка загружается в поле для картинки и при помощи Bitmap.Resize меняется размер картинки на 20x20 пикселей. Полученная картинка переводится в байтовый массив.
3. При нажатии на кнопку «Узнать», картинка полученная во 2 пункте отправляется на сервер через созданный сокет с конечной точкой на локальном сервере(Адрес:127.0.0.1 и Порт:10001)
4. После ожидания ответа сервера выводится: «Самолет», если получена 1,и «Не самолет», если получен 0.

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729. 04.16-01 81 01-1				
Инв. № подл.	Подп. и дата	Взам. Инв. №	Инв. № дубл.	Подп. и дата

3.2.3. Блок работы сервера



0. Создается сокет (Адрес:127.0.0.1 и Порт:10001)
1. Считываются входной поток в массив байт размером 2048. Если размер больше то выбрасывается исключение.
2. Данный массив переводится в RGB изображение размером 20x20 пикселей. Это изображение переводится в черно-белый формат и берется только одн измерения 20x20.
3. Полученный массив передается модели для предсказания, которая возвращает класс модели.
4. Ответ отправляется клиенту

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729. 04.16-01 81 01-1				
Инв. № подл.	Подп. и дата	Взам. Инв. №	Инв. № дубл.	Подп. и дата

3.3. Описание и обоснование выбора метода организации входных и выходных данных

Входные данные:

1. При работе с клиентом: изображения в формате *.png или *.jpg, желательно чтобы фотографии имели размер 20 на 20 пикселей. Если размер другой, то показывается предупреждение о том, что при другом размере ответ может быть неточным
2. Для обучения модели были взяты данные описанные в пункте 3.1.1

Выходные данные: Строка обозначающая класс(“Самолет”, “Не самолет”)

3.4. Описание и обоснование выбора состава технических и программных средств

Состав технических и программных средств, необходимых для работы системы:

1. Серверная часть:
 - Рекомендуется компьютер, оснащенный процессором с частотой 2.7 ГГц или более (четырёх система). Рекомендуется 8ГБ ОЗУ или более. Место на жестком диске: минимум 1 ГБ
 - Клавиатура
 - Установленный Python3 с пакетами: numpy, tensorflow, Pillow, json
 - Операционная система Microsoft Windows Server 8/8.1/10, Linux (Ubuntu 14.04/16.04).
2. Клиентская часть:
 - Мышь или совместимое указывающее устройство и клавиатура.
 - Операционная система Microsoft Windows 8/8.1/10

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729. 04.16-01 81 01-1				
Инв. № подл.	Подп. и дата	Взам. Инв. №	Инв. № дубл.	Подп. и дата

4. ТЕХНИКО-ЭКОНОМИЧЕСКИЕ ПОКАЗАТЕЛИ

4.1. Предполагаемая потребность

Программа может быть использована для автоматизации поиска полных изображений объектов на спутниковых снимках.

4.2. Ориентировочная экономическая эффективность

В рамках данной работы расчёт экономической эффективности не предусмотрен.

4.3. Экономические преимущества разработки по сравнению с отечественными и зарубежными аналогами

На момент начала разработки на рынке не было выявлено аналогичных продуктов

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729. 04.16-01 81 01-1				
Инв. № подл.	Подп. и дата	Взам. Инв. №	Инв. № дубл.	Подп. и дата

5. ПРИЛОЖЕНИЕ

5.1. Используемая литература

1. Руководство по Tensorflow [Электронный ресурс]: Режим доступа <https://www.tensorflow.org/tutorials/> - свободный
2. Документация по языку Python3 [Электронный ресурс]: Режим доступа <https://www.python.org/doc/> - свободный
3. Документация по языку C# [Электронный ресурс]: Режим доступа <https://docs.microsoft.com/ru-ru/dotnet/csharp/programming-guide/> - свободный
4. Дата-сет для тренировки модели [Электронный ресурс]: Режим доступа <https://www.kaggle.com/rhannell/planesnet> - свободный
5. Документация по библиотеке NumPy(Python3) [Электронный ресурс]: Режим доступа <https://docs.scipy.org/doc/numpy/reference/routines.html> - свободный
6. Документация по модулю socket(Python3) [Электронный ресурс]: Режим доступа <https://docs.python.org/3/library/socket.html> - свободный
7. Документация по модулю Socket(System.Net.Sockets)(C#) [Электронный ресурс]: Режим доступа [https://msdn.microsoft.com/ru-ru/library/system.net.sockets.socket\(v=vs.110\).aspx](https://msdn.microsoft.com/ru-ru/library/system.net.sockets.socket(v=vs.110).aspx) - свободный

5.2. Описание и функциональное назначение классов и библиотек

Класс/Библиотека	Назначение
Сервер	
Server	Библиотека сервера
CNN	Свёрточная нейронная сеть
PredProc	Запуск обучения
SupFunc	Вспомогательные функции
Клиент	
MainForm	Главное окно программы

5.3. Описание и функциональное назначение методов, полей и свойств

Server

GetAns	Метод возвращающий результат предсказания
InitServer	Метод инициализирующий сервер
InitNtainedModel	Метод инициализирующий модель если она натренирована
Main	Метод работы сервера

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729. 04.16-01 81 01-1				
Инв. № подл.	Подп. и дата	Взам. Инв. №	Инв. № дубл.	Подп. и дата

CNN

cnn_model_fn	Метод описывающий модель
InitModel	Метод инициализирующий модель
TrainModel	Метод тренировки модели
Predict	Метод предсказывающий по модели

PredProc

main	Метод запускающий тренировку модели
------	-------------------------------------

SupFunc

Kontrast	Функция контрастного фильтра
RGBtogrey	Функция черно-белого фильтра
UnisonShuffle	Метод совместного перемешивания
GetGeyData	Метод переводящий сырые данные в удобные для работы
MakeTrainEvalData	Метод разделяющий данные на тренировочные и проверочные

MainForm

Bitmapbytes	Поле хранящее байтовое представление изображения
BitmapbytesLen	Поле хранящее длину Bitmapbytes
ipAddress	Поле с указанием адреса подключения
remoteEP	Поле с указанием адреса подключения
client	Сокет
MainForm	Конструктор WinForm
buttonLoadImg_Click	Действие происходящее при нажатие на кнопку «Загрузить»
buttonSend_Click	Действие происходящее при нажатии на кнопку «Узнать»
LoadBitmap	Метод загружающий картинку и переводящий её в байтовую строку
resizeImage	Метод изменения размера картинки
GetInfoAboutImg	Метод отправки картинки на сервер и получения ответа

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729. 04.16-01 81 01-1				
Инв. № подл.	Подп. и дата	Взам. Инв. №	Инв. № дубл.	Подп. и дата

