

## **Аннотация**

Данный документ содержит пояснительную записку к программе «Firewatch» (Программа определения лесных пожаров по спутниковым фотографиям с помощью нейронных сетей).

В разделе «Введение» содержится полное и краткое наименование программы и документы, на основании которых ведется разработка.

В разделе «Назначение и область применения» содержатся функциональное и эксплуатационное назначение данной программы.

В разделе «Технические характеристики» содержатся следующие подразделы:

- Постановка задачи на разработку программы
- Описание алгоритма и функционирования программы

В разделе «Технико-экономические показатели» содержится описание предполагаемой потребности в программе, а также определены преимущества данной программы перед отечественными и зарубежными аналогами.

Настоящий документ разработан в соответствии с требованиями:

- 1) ГОСТ 19.101-77 Виды программ и программных документов [1];
- 2) ГОСТ 19.102-77 Стадии разработки [2];
- 3) ГОСТ 19.103-77 Обозначения программ и программных документов [3];
- 4) ГОСТ 19.104-78 Основные надписи [4];
- 5) ГОСТ 19.105-78 Общие требования к программным документам [5];
- 6) ГОСТ 19.106-78 Требования к программным документам, выполненным печатным способом [6];
- 7) ГОСТ 19.404-79 Пояснительная записка. Требования к содержанию и оформлению [7].

## Содержание

1. Введение .....	2
1.1. Наименование программы .....	2
1.2. Основания для разработки .....	2
2. Назначение и область применения .....	3
2.1. Назначение программы .....	3
2.1.1. Функциональное назначение .....	3
2.1.2. Эксплуатационное назначение .....	3
3. Технические характеристики .....	4
3.1. Постановка задачи на разработку программы .....	4
3.2. Описание алгоритма и функционирования программы .....	4
3.2.1. Работа графического приложения .....	5
3.2.2. Отправка запроса на сервер .....	5
3.2.3. Сохранение и загрузка настроек .....	6
3.2.4. Сохранение журналов классификации .....	6
3.2.5. Работа серверной части программы .....	6
3.2.6. Обучение и работа нейронных сетей .....	9
4. Техничко-экономические показатели .....	15
4.1. Предполагаемая потребность .....	15
4.2. Экономические преимущества разработки по сравнению с отечественными и зарубежными аналогами .....	15
5. Список использованных источников .....	16

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.04.09-01 81 01-1				
Инв. № подл.	Подп. и дата	Взам. Инв. №	Инв. № дубл.	Подп. и дата

## 1. Введение

### 1.1.Наименование программы

Наименование программы – «Программа определения лесных пожаров по спутниковым фотографиям с помощью нейронных сетей», наименование на английском – «Software For Detecting Forest Fires In Satellite Images Using Neural Networks», краткое наименование – «Firewatch».

### 1.2.Основания для разработки

Разработка ведется на основании приказа декана ФКН И.В. Аржанцева № 2.3-02/1012-0 2 от 10.12.18.

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.04.09-01 81 01-1				
Инв. № подл.	Подп. и дата	Взам. Инв. №	Инв. № дубл.	Подп. и дата

## 2. Назначение и область применения

### 2.1. Назначение программы

#### 2.1.1. Функциональное назначение

Программа предназначена для обработки спутниковых изображений с целью обнаружения на них лесных пожаров. Изображения классифицируются с помощью одной из двух предусмотренных программой нейронных сетей. Также программой предусмотрено сохранение журналов классификации и сохранение изображений, классифицированных, как пожары, в выбранную пользователем папку.

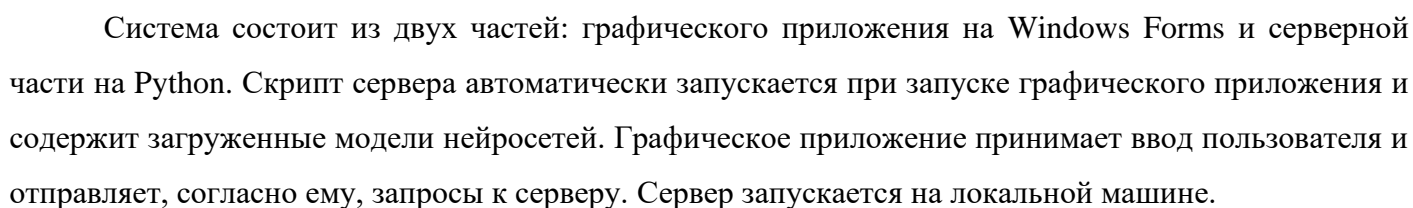
#### 2.1.2. Эксплуатационное назначение

Программа может быть применена для анализа больших объемов спутниковых изображений для определения на них лесных пожаров службами по борьбе с чрезвычайными ситуациями. Автоматическая проверка спутниковых снимков позволяет сократить время, требуемое на их анализ, а также нагрузку лиц, ответственных за эту задачу.

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.04.09-01 81 01-1				
Инв. № подл.	Подп. и дата	Взам. Инв. №	Инв. № дубл.	Подп. и дата

### 3.1. Постановка задачи на разработку программы

### 3.2. Описание алгоритма и функционирования программы



### 3.2.1. Работа графического приложения

Функции графического приложения, направленные на работу с пользователем, отображение элементов интерфейса и вывод сообщений сформированы с помощью элементов Windows Forms и не содержат алгоритмов, нуждающихся в пояснении.

### 3.2.2. Отправка запроса на сервер

Рассмотрим работу графического приложения при отправке запроса на классификацию изображений. По нажатию кнопки Classify Images приложение открывает окно выбора файлов или папок, в зависимости от текущих настроек. После выбора приложение сохраняет список путей к выбранным файлам для последующей передачи на обработку нейросетью. Среди принятых файлов выбираются файлы подходящего для обработки формата путем проверки расширений. При подтверждении выбора пользователем приложение собирает POST-запрос к серверу – объект класса WebRequest. В качестве основной (и единственной) значимой информации запроса отправляется массив байт, содержащий пути к файлам, подлежащим обработке. Конкретнее, список путей собирается в одну строку с символом «\*» в качестве разделителя. Такой выбор разделителя обусловлен тем, что этот символ недопустим в именах файлов и папок операционной системы Windows, на которую рассчитано приложение. Затем полученная строка конвертируется в массив байтов в кодировке utf-8 с помощью объекта Stream. Полученный массив байтов отправляется в POST-запросе на соответствующую страницу по адресу сервера.

Стоит отметить, что можно было бы использовать для отправки информации формат JSON, HTML или какой-либо иной общепринятый формат, однако в этом нет никакой необходимости, потому что как отправляемые, так и получаемые от сервера данные содержат информацию только одного типа – одномерные массивы строк. Масштабирование приложения для интеграции в более крупные системы также не предполагает расширения формата передаваемых и получаемых данных, так как единственная нагрузка сервера – поддержание работы двух предварительно подготовленных нейросетей и обеспечение связи с ними. Для этого достаточно лишь формата списка строк, с которым удобнее работать без попыток обернуть его в структуру типа JSON или HTML.

Выбор путей к файлам, как объекта, отправляемого к серверу, обусловлен тем, что сервер запускается на локальной машине и отправлять в запросе сами файлы было бы нецелесообразно и увеличило бы время работы.

В случае, когда пользователем запрошена загрузка весов для нейросети, запросы с путем к файлу весов отправляются последовательно сначала для модели бинарной классификации, затем, если

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.04.09-01 81 01-1				
Инв. № подл.	Подп. и дата	Взам. Инв. №	Инв. № дубл.	Подп. и дата

загрузка была неудачна, снова для модели, содержащей 6 классов. Если и эта попытка была неудачна, соответствующее сообщение выводится на экран.

### 3.2.3. Сохранение и загрузка настроек

Настройки приложения сохраняются в файл settings.ini в стандартном формате файлов конфигурации – параметры записываются на разных строках, имена и значения разделены символом «=». При загрузке строки файла последовательно считываются, проверяются на соответствие ожидаемым значениям и отображаются на работе программы и в интерфейсе на флажках напротив соответствующих функций. При ошибке считывания файла настройки заполняются значениями по умолчанию. Однако, если файл прочитан, но какие-то из его строк содержат непредвиденные значения, по умолчанию заполняются только функции, содержащие эти значения.

### 3.2.4. Сохранение журналов классификации

Журналы классификации сохраняются в текстовых файлах. Их имена содержат текущую дату и время, когда были получены результаты классификации. В строки файла записываются пути к файлам и классы, данные им нейросетью при классификации. Разделителем между элементами выступает строка “\t:\t”. Такой формат удобен для дальнейшего программного анализа при необходимости.

### 3.2.5. Работа серверной части программы

Серверная часть написана с помощью пакета Flask. При запуске скрипта сервера производится попытка загрузки предварительно обученных моделей нейросетей с весами по умолчанию. Если возникает ошибка при загрузке модели, это отражается в соответствующей переменной. После этого запускается непосредственно сервер, начиная отслеживать поступающие запросы. Если возникла ошибка при загрузке модели или весов, соответствующий ответ будет отправлен при попытке подключения к серверу, программа выведет сообщение для пользователя с информацией об ошибке и будет остановлено. В скрипте сервера описаны функции для обработки запросов, загрузки, предварительной обработки и классификации изображений и загрузки весов.

При поступлении запроса на классификацию путей к изображениям сервер активирует функции загрузки предварительной обработки и классификации для каждого из изображений. Для обработки изображения нейросетью, изображение должно быть загружено в подходящем для обработки формате, то есть в виде тензора 64x64x3. Для этого функция загрузки обращается к соответствующей функции, которая масштабирует изображение. Также эта функция проводит нормализацию изображения, то есть

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.04.09-01 81 01-1				
Инв. № подл.	Подп. и дата	Взам. Инв. №	Инв. № дубл.	Подп. и дата

переводит значения яркости цветов из диапазона  $[0, 255]$  в диапазон  $[0, 1]$ . Таким образом, изображения подготавливаются к процедуре классификации. Классификатор нейросети возвращает одномерный массив чисел с плавающей точкой, отображающий вероятности отнесения данного изображения к каждому из классов. Из полученных значений выбирается наибольшее и сопоставляется соответствующему строковому значению имени класса из заранее определенного массива. Полученный класс добавляется к строке результатов. Если в процессе обработки и классификации изображения возникла ошибка (файл не был найден, имел неподходящий формат или возникла иная проблема), к строке результатов добавляется «Error». Полученная строка результатов отправляется, как ответ на полученный запрос для последующей обработки на клиентской части.

При запросе на загрузку весов происходит попытка обновить веса в соответствии с файлом, находящемся по поступившему пути, для соответствующей нейросети. В процессе этой попытки могут возникнуть ошибки по следующим причинам: файл весов поврежден или сопровождающие его файлы не найдены, веса имеют неверный формат. Например, попытка загрузить веса, подходящие для бинарной модели, в многоклассовую будет неудачна, так как выходной слой первой содержит 2 нейрона, второй – 6. Если попытка загрузки безуспешна, веса остаются прежними, сервер посылает клиенту сообщение об ошибке.

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.04.09-01 81 01-1				
Инв. № подл.	Подп. и дата	Взам. Инв. №	Инв. № дубл.	Подп. и дата



На рис. 1 представлена схема работы основных функций приложения. Для повышения читаемости схемы некоторые функции в ней опущены, как, например: изменение и сохранение настроек, вывод предупредительных сообщений пользователю, запуск и закрытие сервера и прокрутка изображений в окне просмотра.

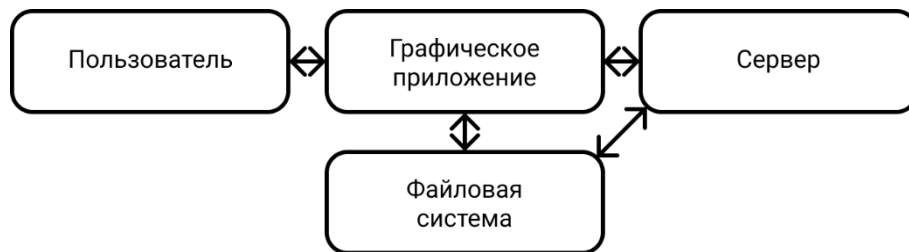


Рисунок 2 – Общая структура связей программы

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.04.09-01 81 01-1				
Инв. № подл.	Подп. и дата	Взам. Инв. №	Инв. № дубл.	Подп. и дата

### 3.2.6. Обучение и работа нейронных сетей

Выбор модели классификации был обусловлен результатами исследовательской работы [8] «SmokeNet: Satellite Smoke Scene Detection Using Convolutional Neural Network with Spatial and Channel-Wise Attention». В ходе исследования нейросеть ResNet-50 показала хороший результат, достигнув более, чем 90-процентной точности на тестовой выборке.

Модель ResNet-50 основана на базовом принципе работы сверточных нейросетей, однако является остаточной сетью. Это отражается в том, что к стандартной архитектуре сверточной сети добавлены соединения быстрого доступа.

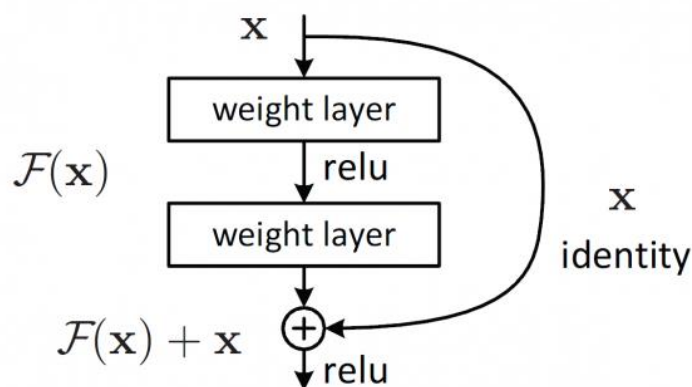


Рисунок 3 – Схема соединения быстрого доступа

Это соединение позволяет пропускать некоторые слои и сопоставлять их входные параметры с выходными. Как было показано в работе [9] «Deep Residual Learning for Image Recognition», такие соединения позволяют решать проблему деградации, когда сети с большим количеством слоев дают точность ниже, чем аналогичные сети с меньшим количеством слоев. Эта проблема кажется контринтуитивной, так как более глубокие сети должны показывать лучшие результаты в аппроксимации нелинейных функций, и решается добавлением соединений быстрого доступа. Данный вывод был получен эмпирически в работе, указанной выше. Таким образом, эти соединения позволяют создавать более глубокие нейросети для более близкой аппроксимации, не страдающие от проблемы деградации. Созданная на их основе модель ResNet стала победителем соревнования по классификации изображений «Large Scale Visual Recognition Challenge 2015 (ILSVRC2015)» [10].

Модель, взятая мной за основу – ResNet-50 – содержит 50 слоев. Вообще, модели ResNet доходят до глубины в 152 слоя, но выбор был сделан в пользу менее глубокой модели ради скорости обучения и работы.

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.04.09-01 81 01-1				
Инв. № подл.	Подп. и дата	Взам. Инв. №	Инв. № дубл.	Подп. и дата

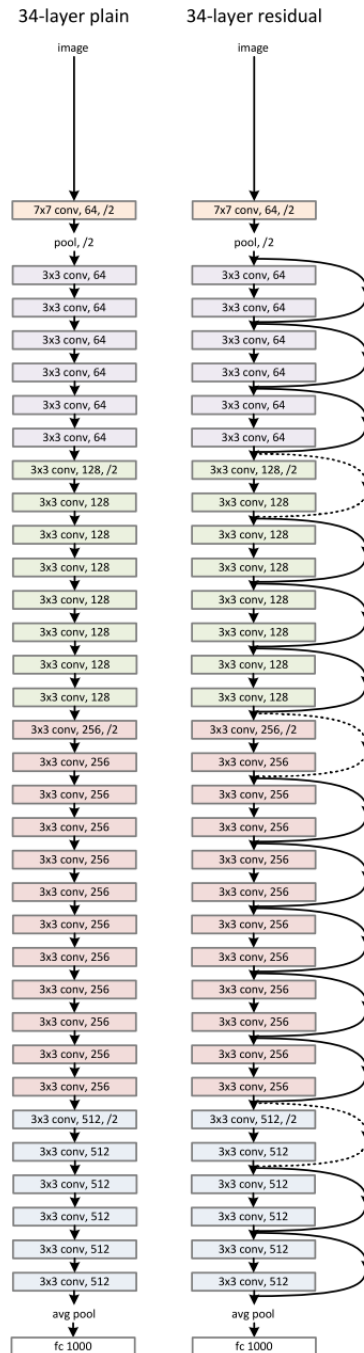


Рисунок 4 – структура модели ResNet [11]

На рис. 4 продемонстрирована разница между структурой обычной сверточной нейросети (слева) и 34-слойной ResNet (справа).

Модель настроена принимать на вход тензоры  $64 \times 64 \times 3$ , представляющие нормализованные RGB-изображения. К структуре ResNet50 добавлен полносвязный слой с функцией активации softmax [12] и 6 выходными параметрами для многоклассовой модели или 2 для бинарной.

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.04.09-01 81 01-1				
Инв. № подл.	Подп. и дата	Взам. Инв. №	Инв. № дубл.	Подп. и дата

Датасет для обучения нейросети был заимствован из работы [8] и является подборкой снимков с датчика MODIS [13], установленного на спутниках систем Terra и Aqua. Снимки были собраны за период с 1999 по 2019 гг. Датасет содержит 6225 изображений, распределенных по 6 классам: облако, пыль, дымка, земля, дым, побережье. Распределение снимков по классам показано ниже:

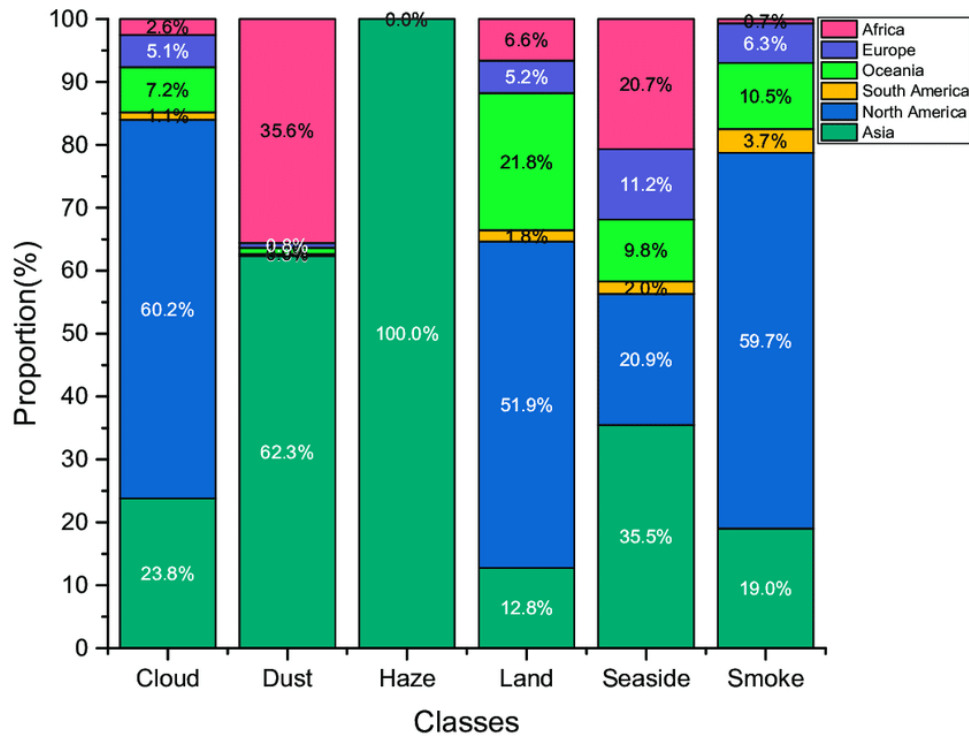


Рисунок 5 – Распределение снимков датасета по классам

Стоит заметить, что, несмотря на то что основной целью классификации является отбор изображений, содержащих лесные пожары, датасет не содержит класса «пожар» или «огонь». Это объясняется тем, что при наблюдении со спутника основным видимым продуктом пожара является именно дым, тогда как огонь зачастую не виден, особенно на начальных этапах распространения пожара, когда его обнаружение со спутника наиболее актуально. Изображения, классов «дымка», «пыль» и «облако» являются негативными классами дыма, а земля и побережье выступают в роли фоновых классов.

Изображения датасета имеют размер 256x256 точек, однако при обучении были преобразованы к размеру 64x64, которое было выбрано эмпирическим путем, так как обеспечивало лучший баланс между точностью за счет сохранения основных элементов изображения и скоростью обработки нейросетью.

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.04.09-01 81 01-1				
Инв. № подл.	Подп. и дата	Взам. Инв. №	Инв. № дубл.	Подп. и дата

Обе нейросети были обучены на зацикленном пакетно-нормализованном датасете, являющемся 70-процентной выборкой исходного в течение 30 эпох, после каждой из которых веса были сохранены. Затем каждое сохранение было протестировано на тестовой выборке, не пересекающейся с тренировочным датасетом. В качестве метрики тестирования была использована метрика «ассигасу» библиотеки Tensorflow. Эта метрика используется на датасете с известными классами и подсчитывает долю предсказаний, совпавших с истинным классом объекта.

```

7
39/39 [=====] - 9s 236ms/step - loss: 0.3676 - accuracy: 0.9407
8
39/39 [=====] - 9s 224ms/step - loss: 0.3669 - accuracy: 0.9471
9
39/39 [=====] - 8s 217ms/step - loss: 0.3834 - accuracy: 0.9295
10
39/39 [=====] - 9s 224ms/step - loss: 0.4149 - accuracy: 0.8942
11
39/39 [=====] - 9s 220ms/step - loss: 0.3673 - accuracy: 0.9471
12
39/39 [=====] - 9s 218ms/step - loss: 0.3621 - accuracy: 0.9487
13
39/39 [=====] - 9s 227ms/step - loss: 0.3638 - accuracy: 0.9471
14
39/39 [=====] - 9s 227ms/step - loss: 0.3656 - accuracy: 0.9423
15
39/39 [=====] - 9s 226ms/step - loss: 0.3569 - accuracy: 0.9567
16
39/39 [=====] - 9s 220ms/step - loss: 0.3690 - accuracy: 0.9439
17
39/39 [=====] - 9s 224ms/step - loss: 0.3706 - accuracy: 0.9375
18
39/39 [=====] - 9s 222ms/step - loss: 0.3505 - accuracy: 0.9615
19
39/39 [=====] - 9s 226ms/step - loss: 0.3601 - accuracy: 0.9487
20
39/39 [=====] - 9s 221ms/step - loss: 0.3743 - accuracy: 0.9343
21
39/39 [=====] - 9s 231ms/step - loss: 0.4290 - accuracy: 0.8782

```

Рисунок 6 – Показатели некоторых эпох обучения на тестовой выборке

Так как результаты большинства эпох оказались очень близки, было проведено второе тестирование на той же выборке. В этом тестировании акцент был сделан на главной задаче нейросети – определении изображений класса «дым». Для этого были выбраны изображения, классифицированные, как дым. Затем эти результаты были сопоставлены с изображениями, действительно содержащими дым. Таким образом, было получено количество верных, ложноположительных и ложноотрицательных предсказаний.

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.04.09-01 81 01-1				
Инв. № подл.	Подп. и дата	Взам. Инв. №	Инв. № дубл.	Подп. и дата

```

Epoch: 7
True Smokes: 89 , False Positives: 25 , False Negatives: 12
Epoch: 8
True Smokes: 79 , False Positives: 11 , False Negatives: 22
Epoch: 9
True Smokes: 78 , False Positives: 21 , False Negatives: 23
Epoch: 10
True Smokes: 90 , False Positives: 54 , False Negatives: 11
Epoch: 11
True Smokes: 70 , False Positives: 2 , False Negatives: 31
Epoch: 12
True Smokes: 75 , False Positives: 6 , False Negatives: 26
Epoch: 13
True Smokes: 77 , False Positives: 9 , False Negatives: 24
Epoch: 14
True Smokes: 78 , False Positives: 12 , False Negatives: 23
Epoch: 15
True Smokes: 85 , False Positives: 11 , False Negatives: 16
Epoch: 16
True Smokes: 67 , False Positives: 1 , False Negatives: 34
Epoch: 17
True Smokes: 80 , False Positives: 18 , False Negatives: 21
Epoch: 18
True Smokes: 88 , False Positives: 11 , False Negatives: 13
Epoch: 19
True Smokes: 89 , False Positives: 20 , False Negatives: 12
Epoch: 20
True Smokes: 73 , False Positives: 13 , False Negatives: 28
Epoch: 21
True Smokes: 89 , False Positives: 63 , False Negatives: 12

```

Рисунок 7 – Показатели тех же эпох во втором тестировании

По результатам тестирования были отобраны 3 сохранения, которые были в дальнейшем использованы в программе: 2 для многоклассовой модели и 1 – для бинарной. Эти веса показали лучшие результаты на тестовой выборке, однако 2 выбранных для многоклассовой модели имели несколько различные показатели.

Файл весов, названный «accurate» - сохранение 18 эпохи обучения, имел наименьшее количество ложноположительных и ложноотрицательных показаний в сумме, тогда как веса эпохи 10 под именем «panicking» выдали больше ложноположительных результатов, но меньше ложноотрицательных, то есть вели себя более «осторожно», предпочитая предсказать наличие пожара там, где его нет, пропуску реального пожара на снимке, который с другими весами не был бы замечен. По большому счету, результаты достаточно схожи в достаточной степени, поэтому выбор в пользу одного набора весов так и не был сделан. Также наличие двух наборов позволяет продемонстрировать возможность программы загружать веса из разных файлов.

По моему личному предположению, бинарная классификация должна была дать лучшие результаты, так как позволяла бы нейросети при обучении «концентрироваться» на параметрах, которые отличают именно класс дыма, однако эта модель дала очень схожие результаты с

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.04.09-01 81 01-1				
Инв. № подл.	Подп. и дата	Взам. Инв. №	Инв. № дубл.	Подп. и дата

многоклассовой (около 93-95% точности на тестовой выборке для обеих моделей), а на тесте определения дыма результаты выбранных весов были практически неотличимы, и было решено оставить для использования в программе обе модели, которые можно применять в зависимости от конкретной поставленной задачи.

Также для дальнейшего применения в работе с пользователем было решено переименовать класс «smoke» (дым) в «wildfire» (пожар) во избежание заблуждения пользователя. Причина, по которой такая замена уместна, описана выше в этом разделе.

По итогам проведенного обучения и тестирования я все еще считаю, что бинарная модель имеет больший потенциал при наличии большего и более нормализованного датасета, но многоклассовая модель может послужить основой для более специфических задач, таких как определение общей климатической обстановки в конкретной области. Веса также могут быть использованы, как основа для обучения на датасетах снимков определенных регионов мира.

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.04.09-01 81 01-1				
Инв. № подл.	Подп. и дата	Взам. Инв. №	Инв. № дубл.	Подп. и дата

## **4. Техничко-экономические показатели**

### **4.1.Предполагаемая потребность**

Программа может быть использована службами по контролю и устранению чрезвычайных ситуаций для раннего обнаружения лесных пожаров. Как было показано в ходе испытаний программы, автоматическая обработка изображений значительно быстрее проверки вручную. Как показывает опыт последних событий, таких как пожары в Австралии 2020 года [14] и в Сибири в 2019 [15], обнаружение пожаров на ранних стадиях чрезвычайно важно для предотвращения крупных катастроф. В этом могут помочь снимки со спутников, обнаруживать пожары на которых позволяет созданная программа.

### **4.2.Экономические преимущества разработки по сравнению с отечественными и зарубежными аналогами**

Аналогов созданной программы в открытом доступе не было обнаружено.

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.04.09-01 81 01-1				
Инв. № подл.	Подп. и дата	Взам. Инв. №	Инв. № дубл.	Подп. и дата



## 5. Список использованных источников

1. ГОСТ 19.101-77 Виды программ и программных документов. //Единая система программной документации. – М.: ИПК Издательство стандартов, 2001.
2. ГОСТ 19.102-77 Стадии разработки. //Единая система программной документации. – М.: ИПК Издательство стандартов, 2001.
3. ГОСТ 19.103-77 Обозначения программ и программных документов. //Единая система программной документации. – М.: ИПК Издательство стандартов, 2001.
4. ГОСТ 19.104-78 Основные надписи. //Единая система программной документации. – М.: ИПК Издательство стандартов, 2001.
5. ГОСТ 19.105-78 Общие требования к программным документам. //Единая система программной документации. – М.: ИПК Издательство стандартов, 2001.
6. ГОСТ 19.106-78 Требования к программным документам, выполненным печатным способом. //Единая система программной документации. – М.: ИПК Издательство стандартов, 2001.
7. ГОСТ 19.404-79 Пояснительная записка. Требования к содержанию и оформлению. //Единая система программной документации. – М.: ИПК Издательство стандартов, 2001.
8. SmokeNet: Satellite Smoke Scene Detection Using Convolutional Neural Network with Spatial and Channel-Wise Attention [Электронный ресурс] – URL: <https://webpages.uncc.edu/cchen62/smokenet.pdf>
9. Deep Residual Learning for Image Recognition [Электронный ресурс] – URL: <https://arxiv.org/pdf/1512.03385.pdf>
10. Large Scale Visual Recognition Challenge 2015 (ILSVRC2015) [Электронный ресурс] – URL: <http://image-net.org/challenges/LSVRC/2015/>
11. ResNet-50 [Электронный ресурс] – URL: <https://www.kaggle.com/keras/resnet50>
12. Activation functions in Neural Networks [Электронный ресурс] – URL: <https://www.geeksforgeeks.org/activation-functions-neural-networks/>
13. Moderate Resolution Imaging Spectroradiometer (MODIS) [Электронный ресурс] – URL: <https://modis.gsfc.nasa.gov/>
14. 2019–20 Australian bushfire season [Электронный ресурс] – URL: [https://en.wikipedia.org/wiki/2019%E2%80%9220\\_Australian\\_bushfire\\_season](https://en.wikipedia.org/wiki/2019%E2%80%9220_Australian_bushfire_season)
15. Лесные пожары в Сибири (2019) [Электронный ресурс] – URL: [https://ru.wikipedia.org/wiki/Лесные\\_пожары\\_в\\_Сибири\\_\(2019\)](https://ru.wikipedia.org/wiki/Лесные_пожары_в_Сибири_(2019))

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.04.09-01 81 01-1				
Инв. № подл.	Подп. и дата	Взам. Инв. №	Инв. № дубл.	Подп. и дата

## Приложение

### Описание и функциональное назначение классов

Таблица 1 – Классы программы

Класс	Назначение
Firewatch	Основной класс приложения, описывающий форму графического приложения. Содержит множество методов, реализующих основные функции сообщения между пользователем и программой.
Sender	Класс, обеспечивающий сообщение графического приложения с серверной частью.
Settings	Класс, описывающий настройки приложения.
Tools	Класс, содержащий методы работы формы с файловой системой, а также некоторые служебные функции.

### Описание и функциональное назначение методов, полей и свойств

Таблица 2 - Методы класса Firewatch

Имя	Мод. доступа	Тип	Аргументы	Назначение
AnalyzePredictions	private	void	-	Анализирует полученные предсказания
BackBut_Click	private	void	object, EventArgs	Обработывает нажатие кнопки Back
ChooseFile	private	bool	-	Выбор файла в диалоговом окне
ChooseFolder	private	bool	string	Выбор папки в диалоговом окне
ChooseImage	private	bool	string	Выбор изображения в диалоговом окне

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.04.09-01 81 01-1				
Инв. № подл.	Подп. и дата	Взам. Инв. №	Инв. № дубл.	Подп. и дата

ChooseSaveFolder	private	bool	string	Выбор папки сохранения в диалоговом окне
ChooseWeights	private	bool	string	Выбор файла весов в диалоговом окне
Connected	private	void	-	Выполняет функции после подключения к серверу
Form1_FormClosing	private	void	object, FormClosingEventArgs	Обработка закрытия формы
HideAll	private	void	-	Скрытие элементов интерфейса основного окна
IfMulticlass_CheckedChanged	private	void	object, EventArgs	Обновление настроек
IfSingle_CheckedChanged	private	void	object, EventArgs	Обновление настроек
LoadImage_Click	private	void	object, EventArgs	Обрабатывает нажатие кнопки Load
Next_Click	private	void	object, EventArgs	Обрабатывает нажатие кнопки Следующее
Prev_Click	private	void	object, EventArgs	Обрабатывает нажатие кнопки Предыдущее
SaveLogs_CheckedChanged	private	void	object, EventArgs	Обновление настроек
SaveSettings_Click	private	void	object, EventArgs	Обрабатывает нажатие кнопки Save Settings
SelectImages_Click	private	void	object, EventArgs	Обрабатывает нажатие кнопки Classify Images
SendPred	private	void	-	Передаёт изображения на классификацию
SendWeights	private	void	string	Передаёт веса на загрузку
ShowImage	private	void	int	Показывает изображение в соответствующем окне
Timer1_Tick	private	void	object, EventArgs	Обрабатывает тик таймера подключения

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.04.09-01 81 01-1				
Инв. № подл.	Подп. и дата	Взам. Инв. №	Инв. № дубл.	Подп. и дата

UploadWeights_Click	private	void	object, EventArgs	Обрабатывает нажатие кнопки Upload Weights
ViewImages_Click	private	void	object, EventArgs	Обрабатывает нажатие кнопки View Images

Таблица 3 - Поля класса Firewatch

Имя	Мод. доступа	Тип	Назначение
attempt	private	int	Текущая попытка подключения к серверу
currentimage	private	int	Индекс текущего изображения при просмотре
filenames	private	List<string>	Пути к файлам
imagepredictions	private	Dictionary<string, string>	Словарь, сопоставляющий файлы с их классами
maxattempt	private	int	Максимальная попытка подключения
pathtoserver	private	string	Путь к скрипту сервера
predictions	private	List<string>	Список полученных предсказаний
server	private	Process	Процесс, отвечающий за сервер
settingspath	private	string	Путь к файлу настроек

Таблица 4 - Методы класса Sender

Имя	Мод. доступа	Тип	Аргументы	Назначение
PredictRequest	public	List<string>	List<string>, bool	Выполняет запрос к серверу на получение предсказаний

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.04.09-01 81 01-1				
Инв. № подл.	Подп. и дата	Взам. Инв. №	Инв. № дубл.	Подп. и дата

Replace	public	List<string>	List<string>	Форматирует строки путей к файлам для отправки на сервер
TryToConnect	public	string	-	Выполняет попытку подключения к серверу
WeightsRequest	public	bool	string	Выполняет запрос к серверу на загрузку весов

Таблица 5 - Поля класса Sender

Имя	Мод. доступа	Тип	Назначение
uri	private	string	URI адрес сервера

Таблица 6 - Методы класса Settings

Имя	Мод. доступа	Тип	Аргументы	Назначение
LoadSettings	public	bool	string	Выполняет загрузку настроек
SaveSettings	public	bool	string	Выполняет сохранение настроек

Таблица 7 - Свойства класса Settings

Имя	Мод. доступа	Тип	Назначение
IfMulticlass	public	bool	Какую модель использовать для классификации
IfSingle	public	bool	Загружать отдельные файлы или папки
SaveLogs	public	bool	Сохранять ли журналы классификации

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.04.09-01 81 01-1				
Инв. № подл.	Подп. и дата	Взам. Инв. №	Инв. № дубл.	Подп. и дата

Таблица 8 - Методы класса Tools

Имя	Мод. доступа	Тип	Аргументы	Назначение
GetAndRemove	public	string	List<string>, string	Получает файлы из директории, форматирует их имена и убирает неподдерживаемые файлы
GetClass	public	string	String, Dictionary<string, string>	Получает класс изображения из предсказаний
RemoveUnsupported	public	bool	List<string>	Убирает неподдерживаемые файлы
SaveImages	public	bool	string, Dictionary<string, string>	Сохраняет изображения в папку
SaveLogs	public	bool	Dictionary<string, string>	Сохраняет журнал классификации
StartServer	public	void	Process, string	Запускает сервер

Таблица 9 - Свойства класса Tools

Имя	Мод. доступа	Тип	Назначение
AllowedFormats	public	string[]	Разрешенные форматы изображений

### Описание функций скрипта server.py

Имя	Тип	Аргументы	Назначение
index	str	-	Возвращает сообщение, отражающее статус сервера
load_and_preprocess_image	tensorflow.python.framework.ops.EagerTensor	str	Загружает файл и выполняет предобработку

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.04.09-01 81 01-1				
Инв. № подл.	Подп. и дата	Взам. Инв. №	Инв. № дубл.	Подп. и дата

make_binary_prediction	str	str	Получает предсказание бинарной модели по файлу
make_multi_prediction	str	str	Получает предсказание многоклассовой модели по файлу
predictmanybinary	str	-	Возвращает предсказания бинарной модели для полученных изображений
predictmanymulti	str	-	Возвращает предсказания многоклассовой модели для полученных изображений
preprocess_image	tensorflow.python.- framework.ops.EagerTensor	tensorflow.python.- framework.ops.EagerTensor	Выполняет предобработку изображения
uploadweightsbinary	str	-	Загружает веса для бинарной модели
uploadweightsmulti	str	-	Загружает веса для многоклассовой модели

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.04.09-01 81 01-1				
Инв. № подл.	Подп. и дата	Взам. Инв. №	Инв. № дубл.	Подп. и дата

## Лист регистрации изменений

[illegible]

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.04.09-01 81 01-1				
Инв. № подл.	Подп. и дата	Взам. Инв. №	Инв. № дубл.	Подп. и дата