

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ
РОССИЙСКОЙ ФЕДЕРАЦИИ
Федеральное государственное автономное образовательное учреждение
высшего образования
**«Южно-Уральский государственный университет
(национальный исследовательский университет)»**
Высшая школа электроники и компьютерных наук
Кафедра системного программирования

РАБОТА ПРОВЕРЕНА

Рецензент

Технический директор

ООО «Компания СГ-групп»

_____ Д.В.Гапонюк

“ ____ ” _____ 2019 г.

ДОПУСТИТЬ К ЗАЩИТЕ

Заведующий кафедрой, д.ф.-м.н.,

профессор

_____ Л.Б. Соколинский

“ ____ ” _____ 2019 г.

**РАЗРАБОТКА ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ
ДЛЯ ВЫЯВЛЕНИЯ ЛЕСНОГО ПОЖАРА ПО ДАННЫМ
АЭРОФОТОСЪЕМКИ С ПОМОЩЬЮ НЕЙРОСЕТЕВЫХ
ТЕХНОЛОГИЙ**

ВЫПУСКНАЯ КВАЛИФИКАЦИОННАЯ РАБОТА
ЮУрГУ – 09.03.02.2019.115-003.ВКР

Научный руководитель,
к.ф.-м.н., доцент кафедры СП

_____ В.А. Голодов

Автор работы,
студентка группы КЭ-405

_____ А.А. Бурая

Ученый секретарь
(нормоконтролер)

_____ О.Н. Иванова

“ ____ ” _____ 2019 г.

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ
РОССИЙСКОЙ ФЕДЕРАЦИИ
Федеральное государственное автономное образовательное учреждение
высшего образования
**«Южно-Уральский государственный университет
(национальный исследовательский университет)»**
Высшая школа электроники и компьютерных наук
Кафедра системного программирования

УТВЕРЖДАЮ
Зав. кафедрой СП

_____ Л.Б. Соколинский
09.02.2019

ЗАДАНИЕ

на выполнение выпускной квалификационной работы бакалавра
студентке группы КЭ-405
Бурой Александре Анатольевне,
обучающейся по направлению
09.03.02 «Информационные системы и технологии»

- 1. Тема работы** (утверждена приказом ректора от 05.04.2019 № 899)
Разработка программного обеспечения для выявления лесного пожара по данным аэрофотосъемки с помощью нейросетевых технологий.
- 2. Срок сдачи студентом законченной работы:** 05.06.2019.
- 3. Исходные данные к работе**
 - 3.1. База данных Corsican Fire Database (<http://cfdb.univ-corse.fr>)
 - 3.2. Саймон Хайкин, «Нейронные сети: полный курс, 2-е издание», 2006.
- 4. Перечень подлежащих разработке вопросов**
 - 4.1. Провести анализ программных аналогов и научной литературы в предметной области.
 - 4.2. Подготовить обучающую и тестовую выборки.
 - 4.3. Разработать нейронную сеть для распознавания пожара.
 - 4.4. Обучить и протестировать нейронную сеть.
 - 4.5. Реализовать систему выявления лесного пожара, состоящую из клиентского приложения и серверной части.
 - 4.6. Протестировать работу системы.
- 5. Дата выдачи задания:** 09.02.2019.

Научный руководитель
к.ф.-м.н., доцент кафедры СП
Задание принял к исполнению

В.А. Голодов
А.А. Бурая

ОГЛАВЛЕНИЕ

ВВЕДЕНИЕ	4
1. АНАЛИЗ ПРЕДМЕТНОЙ ОБЛАСТИ	8
1.1. Изучение предметной области.....	8
1.2. Обзор аналогичных проектов и существующих решений	9
1.3. Обзор готовых решений для создания нейронных сетей	12
1.4. Применение нейронных сетей	13
2. ТЕОРЕТИЧЕСКАЯ ЧАСТЬ.....	15
2.1. Предобработка входных данных	15
2.2. Принцип работы сверточных нейронных сетей	15
3. ПРОЕКТИРОВАНИЕ	18
3.1. Проектирование работы системы.....	18
3.2. Проектирование топологии нейронной сети	20
3.3. Требования к системе.....	20
3.4. Варианты использования системы	21
4. РЕАЛИЗАЦИЯ	24
4.1. Программные средства реализации.....	24
4.2. Реализация нейронной сети	25
4.3. Реализация серверной части системы	26
4.4. Реализация клиентской части системы	27
5. ТЕСТИРОВАНИЕ	30
5.1. Результаты тестирования	30
5.1.1. Тестирование нейронной сети	30
5.1.2. Функциональное тестирование.....	30
ЗАКЛЮЧЕНИЕ	33
ЛИТЕРАТУРА.....	35
ПРИЛОЖЕНИЯ.....	38
Приложение 1	38
Приложение 2	39

ВВЕДЕНИЕ

ОСНОВНЫЕ ОПРЕДЕЛЕНИЯ

Нейронная сеть – математическая модель, а также ее программная или аппаратная реализация, построенная по принципам организации и функционирования биологических нейронных сетей – сетей нервных клеток живого организма [28].

Синапсы – связь между двумя нейронами, которая имеет единственный параметр – вес [25].

Искусственный нейрон – узел искусственной нейронной сети, являющийся упрощенной моделью естественного нейрона [32].

Классификация – один из разделов машинного обучения, посвященный решению следующей задачи. Имеется множество объектов (ситуаций), разделенных некоторым образом на классы. Задано конечное множество объектов, для которых известно, к каким классам они относятся. Это множество называется обучающей выборкой. Классовая принадлежность остальных объектов не известна. Требуется построить алгоритм, способный классифицировать произвольный объект из исходного множества [21].

Машинное обучение – класс методов искусственного интеллекта, характерной чертой, которых является не прямое решение задачи, а обучение в процессе применения решению множества сходных задач [17].

АКТУАЛЬНОСТЬ ТЕМЫ

Среди множества вариантов атак на лес пожар является одним из самых опасных и неконтролируемых бедствий, с которым каждый год сталкиваются службы спасения и сотни волонтеров. Так, например, 27.04.18 в лесах Южного Урала было ликвидировано порядка 18 возгораний в разных лесничествах. В тушении пожаров было задействовано 176 человек и 60 единиц техники. Выгорело примерно 150,9 гектара земли [23].

Лесные пожары представляют большой риск для людей, которые живут в лесу или с ним по соседству и их здоровья, например, для людей, имеющих

заболевания органов дыхания. Ежегодно вследствие лесных пожаров сотни людей погибают или несут материальный ущерб. Кроме того, гибнет огромное количество диких животных. В случае если лес выгорает полностью, сменяется видовой состав флоры и фауны, а для полного восстановления леса после пожара требуются не один десяток лет. Из-за сокращения лесов меняется кислородный баланс в атмосфере, водный режим рек, озер, болот, они пересыхают или наоборот разливаются. Во время пожаров в атмосферу выбрасывается углекислый газ, дым и пепел.

Несмотря на важность этой проблемы, она остается нерешенной на данный момент. Каждый год на территории всех областей проводится патрулирование (мониторинг обстановки) лесных массивов, однако зачастую, возгорание леса возникает случайно и в труднодоступных для наблюдения местах, что существенно усложняет процесс его устранения.

Наиболее актуальной задачей в комплексе мероприятий по предотвращению лесных пожаров является быстрое обнаружение очага возгорания. Это является одним из главных условий эффективного ведения лесного хозяйства и сохранения его экологического потенциала, а также позволяет вовремя принять необходимые меры безопасности. Россия является крупнейшим источником леса, на нее приходится больше половины хвойных и треть всех лесов нашей планеты. Именно поэтому проблема лесных пожаров является одной из наиболее важных национальных задач.

На данный момент существуют различные устройства выявления лесных пожаров, основной принцип которых это детекция дыма. Однако такие приборы характеризуются большим числом ложных тревог. И даже при использовании очень качественного оборудования высок риск ложного срабатывания.

Инфракрасные камеры или тепловизоры способны засечь области с повышенной температурой не только ночью, но и при сильном задымлении, что делает их весьма эффективными в борьбе с пожарами.

Так же для решения поставленной задачи предлагается использование сверточных нейронных сетей. На данный момент, нейросетевые технологии достигли высокого уровня развития и уже позволяют находить решение специфическим задачам, работая по принципу, похожему на процесс принятия решения человеком. Использование таких методов решения на данном этапе развития технологий не может сделать принятие решений полностью автоматизированным, однако оно существенно снижает влияние человеческого фактора и обеспечивает высокое качество результата при меньших затратах времени и усилий.

ЦЕЛЬ И ЗАДАЧИ ИССЛЕДОВАНИЯ

Целью данной работы является разработка программного обеспечения для выявления лесного пожара по данным аэрофотосъемки с помощью нейросетевых технологий. Для достижения поставленной цели необходимо решить следующие задачи

1. Провести анализ программных аналогов и научной литературы в предметной области.
2. Подготовить обучающую и тестовую выборки.
3. Разработать нейронную сеть для распознавания пожара.
4. Обучить и протестировать нейронную сеть.
5. Реализовать систему выявления лесного пожара, состоящую из клиентского приложения и серверной части.
6. Протестировать работу системы.

СТРУКТУРА И ОБЪЕМ РАБОТЫ

Работа состоит из введения, пяти глав, заключения, библиографии и 2 приложений. Объем работы составляет 37 страниц, объем библиографии – 32 источника, объем приложений – 2 страницы.

СОДЕРЖАНИЕ РАБОТЫ

В первой главе «Анализ предметной области» рассмотрены исследования по использованию нейросетевых технологий в области, в

рамках которой выполняется данная работа и обзор уже существующих решений поставленной задачи.

Вторая глава «Теоретическая часть» содержит подробное описание алгоритмов, применяемых в решении поставленной задачи.

Третья глава «Проектирование» содержит общее описание архитектуры системы, описание этой системы компонентов.

В четвертой главе «Реализация системы» приводится техническая реализация системы исходя из поставленного списка требований, а также описание подобранной топологии нейронной сети. Описан процесс обучения нейронной сети.

В пятой главе «Тестирование» разработан и проведен набор тестов для функционального тестирования системы.

В заключении приводятся основные результаты выполненной работы.

Приложения содержат дополнительные материалы работы: листинги реализованных модулей системы.

1. АНАЛИЗ ПРЕДМЕТНОЙ ОБЛАСТИ

1.1. Изучение предметной области

Пожар – это неконтролируемое горение, уничтожающее или повреждающее материальные ценности, создающее опасность для жизни людей [24]. В любой ситуации даже небольшое возгорание может перерасти в стихийное бедствие. В настоящее время вероятность возгорания и масштабного распространения огня из-за природных факторов не превышает 20 %. Большинство лесных пожаров спровоцировано деятельностью людей [26].

Наиболее распространенные причины возникновения пожара в лесу [22]:

- 1) безответственное поведение людей;
- 2) детская шалость с огнем;
- 3) сжигание мусора вблизи жилых домов и на территории, прилегающей к лесным массивам;
- 4) удар молнии;
- 5) поджог.

Инфракрасное излучение – электромагнитное излучение, которое занимает спектральную область между красным светом и коротковолновым радиоизлучением [19]. Впервые оно было обнаружено в 1800 году астрономом Уильямом Гершелем. Его иногда называют «тепловым излучением», поскольку инфракрасное излучение от нагретых тел воспринимается кожей человека как ощущение тепла. Такое излучение испускают все тела, температура которых отличается от нуля.

Инфракрасное излучение широко используется в промышленности, научных исследованиях и медицине. В этой области, существует огромное количество разработок, основанных на электромагнитном излучении, например: приборы для наблюдения в темноте, приборы для дистанционного измерения температуры тел и приборы для обнаружения целей по их собственному тепловому инфракрасному излучению (термография). Эти

приборы преобразовывают окружающий фотон света в электроны, которые усиливают химические и электрические процессы и преобразуются обратно в видимый источник свет.

Использование данных в тепловом инфракрасном фильтре позволяет избежать работы с задымлением (рис. 1) и увеличить точность работы алгоритма распознавания, снизив процент ошибки.



Рис. 1. Оригинальное изображение лесного пожара и в инфракрасном спектре

1.2. Обзор аналогичных проектов и существующих решений

«FireWatch» [3]

Автоматизированная система обнаружения лесных пожаров «FireWatch». Система работает путем распознавания облаков дыма. Каждые 6 минут сенсор (рис. 2), установленный на высоте 10 метров над лесным покровом, совершает оборот в 360 градусов и фиксирует наличие дыма. В случае нахождения признаков задымления, сигнал тревоги отправляется через беспроводное соединение в центр управления, где оператор принимает решение о дальнейших действиях (рис. 3).

Основными недостатками этой системы является: дороговизна установки, ограниченность обзора (наличие слепых зон) и сильная зависимость результата работы от погодных условий. На некоторых территориях нередким явлением является туман, это означает, что система будет работать некорректно.



Рис. 2. Пример сенсора системы FireWatch

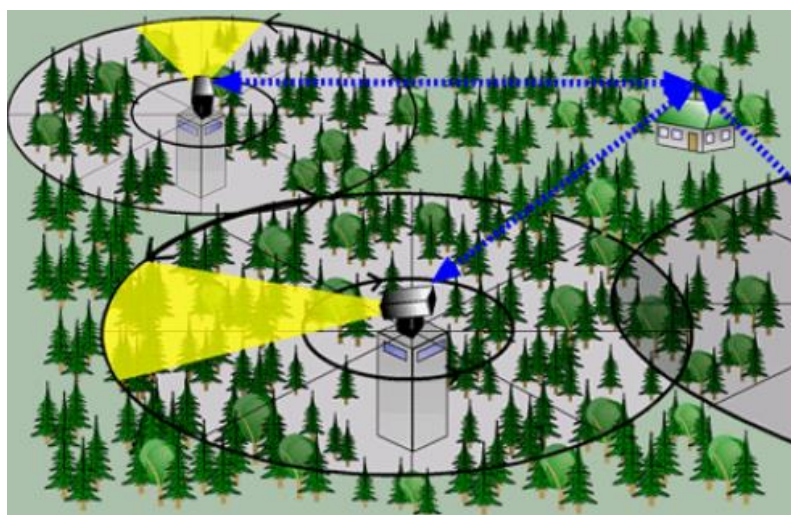


Рис. 3. Принцип работы системы FireWatch

Forest Fire Detection Using Artificial Neural Network Algorithm Implemented in Wireless Sensor Networks [16]

Юншен Лю, Янсун Ян, Чанг Лю и Юй Гу из Исследовательского института сетевых технологий в Китае предложили систему обнаружения лесных пожаров, основанную на искусственной нейронной сети и реализованную с помощью беспроводной сенсорной сети (Wireless Sensor Networks или WSN). Ученые полагают, что использование системы, которая зависит только от одного критерия лесного пожара, увеличивает вероятность ложных срабатываний, поэтому их алгоритм учитывает несколько атрибутов

(многокритериальное обнаружение). Хотя случались и ложные тревоги. Главной причиной их возникновения была чувствительность фотоэлектрического блока обнаружения дыма к белым частицам от тлеющего огня и пыли.

Модуль обнаружения пожара отвечает за принятие решения и отправке сигнала тревоги. Он содержит пять сенсорных узлов, четыре из которых собирают данные о возможном возникновении пожара. Нейронная сеть реализована на каждом отдельном сенсорном узле. Пятый узел датчика работает, как базовая станция и собирает данные зондирования и пожарную тревогу от других четырех узлов, используя беспроводную связь.

Для того чтобы непрерывно использовать систему датчиков было принято решение об установке солнечных батарей потому, что это самый дешевый и экологичный вариант.

Поскольку в лесу доступен только прерывистый солнечный свет, учеными была разработана специальная модель солнечной батареи. Энергия солнечной панели накапливается суперконденсатором.

Модуль пользовательского интерфейса отвечает за отображение данных, считанных с датчиков. Данные зондирования и пожарная сигнализация передаются от базовой станции к пользователю с помощью интернета, поэтому пользователь может находиться вдали от системы пожаротушения.

Image-Based Fire Detection Using Neural Networks [5]

Wen-Bing Horng и Jian-Wen Peng из Тайваньского университета (факультет компьютерных наук и инженерии) разработали алгоритм обнаружения огня на основе изображений с использованием нейронных сетей обратного распространения. Этот метод состоит из 3 этапов: сегментация модели цветного пламени, удаление ложных областей, которые похожи на пожар и оценка пожароопасности. На первом этапе три компонента HSI модели (тон, насыщенность и светлота) каждого пикселя подаются на обучение нейронной сети. Если пиксель не относится к

обнаруженной области пламени, он закрашивается черным, в противном случае цвет не меняется. Примеры работы этого этапа приведены на рис. 4.

Ложными полями может быть отражение пламени и объекты с похожими цветами. В первой ситуации область можно отфильтровать путем удаления пикселей с наименьшей интенсивностью. Во втором случае учитывается цвет фона объекта. Если он аналогичен или очень похож, то метод разности удалит объект с изображения.



Рис. 4. Изображение в оригинале и после обработки нейронной сетью

1.3. Обзор готовых решений для создания нейронных сетей

Для языка программирования Python существует большое количество готовых библиотек для работы с нейронными сетями. Использование таких средств призвано облегчить процесс разработки нейронных сетей.

TensorFlow [15]

Открытая программная библиотека для машинного обучения, разработанная компанией Google для решения задач построения и тренировки нейронной сети с целью автоматического нахождения и классификации образов, достигая качества человеческого восприятия [29].

Keras [7]

Высокоуровневый программный интерфейс, нацеленный на оперативную работу с сетями глубокого обучения, при этом спроектированный так, чтобы быть компактным и расширяемым [30].

OpenCV [11]

Библиотека алгоритмов компьютерного зрения, обработки изображений и численных алгоритмов общего назначения с открытым кодом [18].

1.4. Применение нейронных сетей

Нейронная сеть представляет собой систему программного обеспечения, созданную по образцу работы нейронов в человеческом мозге.

Классификация изображений было одной из первых областей, к которым были успешно применены искусственные нейронные сети (ИНС), однако они успешно применяются и в других областях (робототехника, связь, интернет, планирование и оптимизация процессов). Основным критерием использования ИНС является наличие процесса, который работает в соответствии со строгими правилами или шаблонами и имеет большой объем данных. Если задействованные данные слишком велики, чтобы человек мог понять их в разумные сроки, этот процесс, скорее всего, является главным кандидатом для автоматизации с помощью нейронных сетей.

Нейронные сети невероятно универсальны, пластичны, могут распознавать уникальные закономерности и имеют способность выдавать обоснованный результат, основываясь на данные, которых не было получено в процессе обучения. Распознавание образов является одной из основных областей применения нейросетевых технологий.

Производительность таких алгоритмов напрямую зависит от количества нейронов в каждом слое, включая входной. Например, для последовательности изображений разрешения 64x64 необходим вектор с 4096 пикселями. Процесс подачи такого количества признаков на вектор нейронной сети повлечет за собой слишком большое время работы. Эту проблему можно решить с помощью использования сверточных нейронных сетей. Обычно сети именно такой архитектуры используются при распознавании лиц, оцифровки текста и обработки изображений, поскольку имеют весомое практическое преимущество. Оно заключается в том, что

меньшее количество параметров значительно сокращает время, которое необходимо потратить на обучение, а также уменьшает объем данных, необходимых для обучения модели.

Выводы по первой главе

В соответствии с целями данной работы был проведен обзор научной литературы и существующих аналогов. Наличие аналогов реализованных по разным технологиям подтверждает актуальность поставленной задачи. В ходе обзора были рассмотрены основные недостатки существующих решений.

В результате обзора существующих библиотек и программных интерфейсов было принято решение использовать Keras, по причине простоты в использовании, а также функционала позволяющего построить необходимую нейронную сеть.

В ходе анализа была сформулирована следующая задача: выделить пятно повышенной температуры на изображении и проанализировать полученную информацию с помощью сверточной нейронной сети. На вход будет подаваться изображение в инфракрасном спектре. Выход нейронной сети представляет собой два вещественных числа: 2 (отсутствие пожара на изображении) и 1 (система определила наличие огня).

2. ТЕОРЕТИЧЕСКАЯ ЧАСТЬ

2.1. Предобработка входных данных

Для корректной работы обученной модели нейронной сети необходимо входные данные программы привести в вид, соответствующий заданным при обучении требованиям. На вход системе распознавания лесных пожаров подается изображение любого размера, полученное с камер инфракрасного спектра. Согласно требованиям, нейронная сеть должна получать на вход черно-белое изображение размера 128x72 пикселя.

Задача предобработки входных данных представляется в виде двух последовательных шагов.

Первый шаг заключается в изменении размеров изображения.

Второй шаг заключается в переводе изображения в черно-белую цветовую гамму.

Функционал общедоступной библиотеки OpenCV полностью удовлетворяет всем требованиям задачи предобработки входных данных.

2.2. Принцип работы сверточных нейронных сетей

Каждое изображение состоит из пикселей, который представлен числом от 0 до 255. Поэтому каждое изображение имеет цифровое представление, позволяющее компьютеру работать с ним.

Архитектура сверточных нейронных сетей (рис. 5) впервые была предложена, французским ученым в области информатики, Яном Лекуном. Основной идеей работы таких нейронных сетей является чередование слоев свертки и слоев подвыборки.

Первым слоем ИНС является входной. После него всегда идет слой свертки. Его алгоритм работы состоит из следующего: каждый фрагмент обрабатываемого изображения умножается на матрицу свертки, затем полученные результаты суммируются и записываются в исходное положение на другом изображении. Этот процесс изображен на рис. 5 [10]. Веса ядер задаются случайными числами (из диапазона от -0,5 до 0,5).

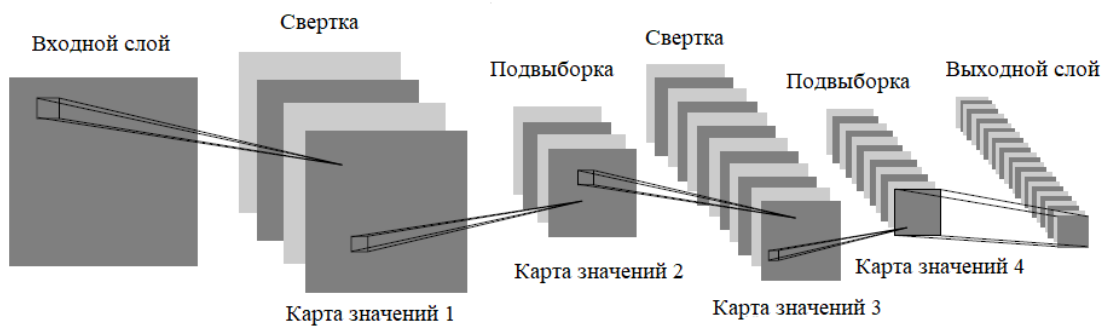


Рис. 5. Пример сверточной нейронной сети

Кроме слоев свертки, используются слои подвыборки. Они позволяют уменьшить размерность изображения. Ключевая особенность метода заключается в том, что группа пикселей (например, 3x3) сжимается до одного. Обычно для этого используется функция максимума. Пример работы слоя подвыборки представлен на рис. 6.

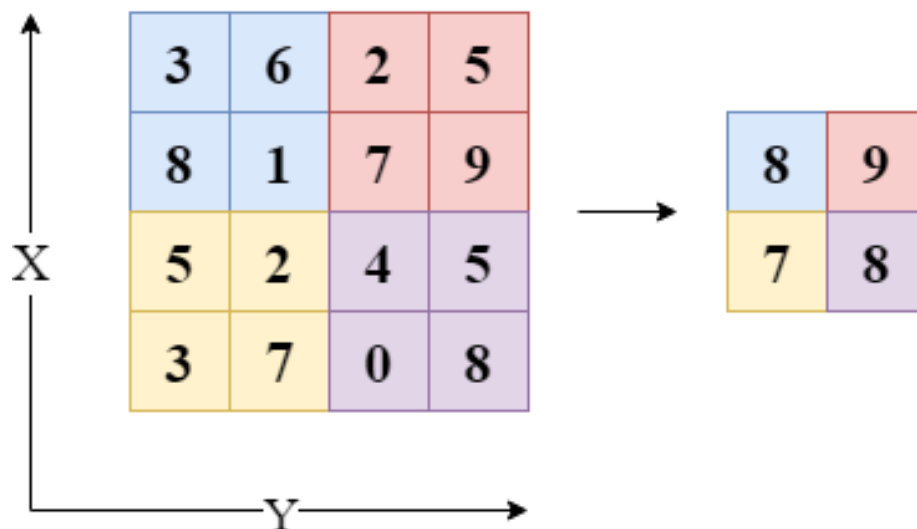


Рис. 6. Пример работы слоя подвыборки с функцией максимума

В результате прохождения по нескольким слоям свертки и подвыборки, количество каналов увеличивается, а размерность изображения уменьшается. В результате, последний слой имеет размерность, соответствующую числу классов.

Выводы по второй главе

В этой главе была рассмотрена теоретическая часть сверточных нейронных сетей, а именно принципы их работы, информация о слоях свертки и подвыборки. Был сформулирован процесс предобработки изображения.

3. ПРОЕКТИРОВАНИЕ

Для решения задачи распознавания пожара в лесу должна быть разработана система, обеспечивающая возможность автоматизированного определения большого количество изображений и состоящая из клиентского приложения и серверной части.

3.1. Проектирование работы системы

Целью работы системы является выявление лесных пожаров на данных, записанных с квадрокоптеров в инфракрасном спектре. Квадрокоптеры летают на лесным массивом и периодически возвращаются на базу дозарядиться. На базе находятся технические специалисты, обслуживающие квадрокоптер и выполняющие взаимодействие с системой распознавания пожаров на изображении (непосредственно загружают файлы для дальнейшего анализа с целью своевременного выявления очагов возгорания лесной зоны). В ходе проектирования были рассмотрены два варианта реализации системы распознавания пожара.

Первый вариант заключается в анализе данных с квадрокоптера непосредственно на серверах базы. В этом случае обученная модель нейронной сети расположена на клиентской части приложения. К достоинствам данного варианта можно отнести автономность системы.

Второй вариант заключается в анализе данных с квадрокоптера на удаленных серверах. Данные серверов содержат обученную модель нейронной сети. Сервера общедоступны, любая база может отправить на нее свои файлы и получить вердикт о наличии пожара в качестве ответа. К достоинствам данного варианта можно отнести слабую загруженность клиентской части приложения и повсеместный доступ к наиболее актуальной версии аналитической части программы, расположенной на сервере.

Исходя из превосходящих преимуществ второго варианта было принято решение взять его за основу в рамках данной работы.

Проектирование клиентской части

Клиентская часть должна состоять из интерфейса для удобного взаимодействия с системой, программного кода алгоритмов загрузки и модуля, ответственного за взаимодействие с сервером (отправку данных и получение ответа).

После выбора пользователем директории с изображениями для анализа наличия на них пожара и нажатия кнопки «Analyze» начинается выполнение загрузки выбранных файлов в программу и их отправкой на сервер. Все файлы из выбранной пользователем директории с подходящим расширением (.jpg и .png) поочередно загружаются в программу и помещаются в json форму для дальнейшей отправки на сервер. Взаимодействие между клиентом и сервером осуществляется по протоколу REST API в формате json. Клиент отправляет на сервер выбранное изображение (файл). Сервер принимает данное сообщение, осуществляет предварительную обработку полученного изображения, заключающуюся в изменении размера изображения и переводе в черно-белую цветовую гамму, анализирует, полученное в результате предобработки, изображение при помощи обученной модели нейронной сети и отправляет результат в виде вердикта, вынесенного нейронной сетью типа boolean (True или False) и наименование обработанного изображения. Далее на клиенте обрабатываются данные, полученные с сервера. Наименование изображения и оценка пожароопасности заносятся в таблицу результатов анализа. В том случае, если клиент получил вердикт False в столбце «results» записывается «No». Если клиент получил вердикт True в столбце «results» записывается «Yes» и наименование изображения с положительным вердиктом заносится в поле-селектор под таблицей. При выборе в данном селекторе наименования какого-либо файла и последующего нажатия кнопки «SHOW» будет запущено дополнительное окно с возможностью просмотра необработанного изображения, соответствующего выбранному наименованию.

Проектирование серверной части

Серверная часть приложения состоит из программного кода алгоритмов взаимодействия с клиентской частью (получения и отправки данных в формате json) по протоколу REST API, функции аналитической обработки изображения заранее обученной моделью нейронной сети, ответственной за оценку пожароопасности, и алгоритма предварительной обработки полученных с клиентской части данных, загруженных пользователем.

3.2. Проектирование топологии нейронной сети

Для решения поставленной задачи была разработана топология сверточной нейронной сети, код которой представлена на рис. 7.

Данная топология содержит 3 сверточных слоя, 3 слоя подвыборки и 2 полносвязных слоя. Представленная выше топология выбрана экспериментальным путем.

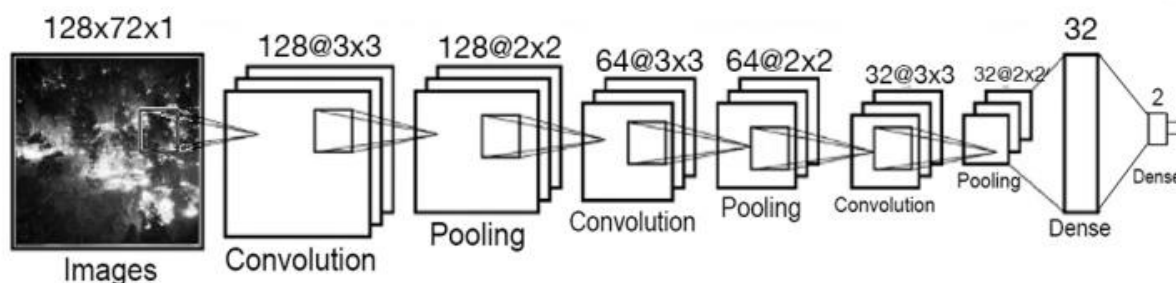


Рис. 7. Топология нейронной сети

3.3. Требования к системе

Функциональные требования описывают поведение системы, т.е. ее действия (вычисления, преобразования, проверки, обработку и т.д.) [20].

Разрабатываемая система должна удовлетворять следующим функциональным требованиям.

1. Система должна принимать на вход изображение в инфракрасном спектре.
2. Система должна выявить возможное наличие огня на изображении.
3. Система должна вывести на экран сообщение с результатами работы.
4. Система должна вывести на экран изображение, на котором был обнаружен огонь.

Нефункциональные требования описывают свойства системы (удобство использования, безопасность, надежность, расширяемость и т.д.), которыми она должна обладать при реализации своего поведения [20].

Поскольку существуют большое количество готовых библиотек и надстроек для создания нейронных сетей, система должна удовлетворять следующим нефункциональным требованиям.

1. Система должна быть реализована на языке Python 3.7.
2. Система должна использовать фреймворк для создания нейронных сетей Keras.
3. Система должна быть единой и не требовать от пользователя дополнительных действий для работы, кроме запуска и указания входных данных и параметров.

3.4. Варианты использования системы

Для проектирования системы был использован язык графического описания для объектно-ориентированного программирования UML. На рис. 8. представлена построенная модель взаимодействия актера «Пользователь» с системой распознавания в виде диаграммы вариантов использования (use-case diagram) [27].

Актеры, взаимодействующие с системой:

Пользователь – человек, использующий систему и ее определенные функции.

Краткое описание вариантов использования:

Пользователю доступны следующие функции:

- 1) выбрать директорию с изображениями;
- 2) запустить процесс классификации загруженных файлов;
- 3) посмотреть проанализированные изображения с положительным вердиктом.

Диаграмма деятельности

На основе функциональных и нефункциональных требований была

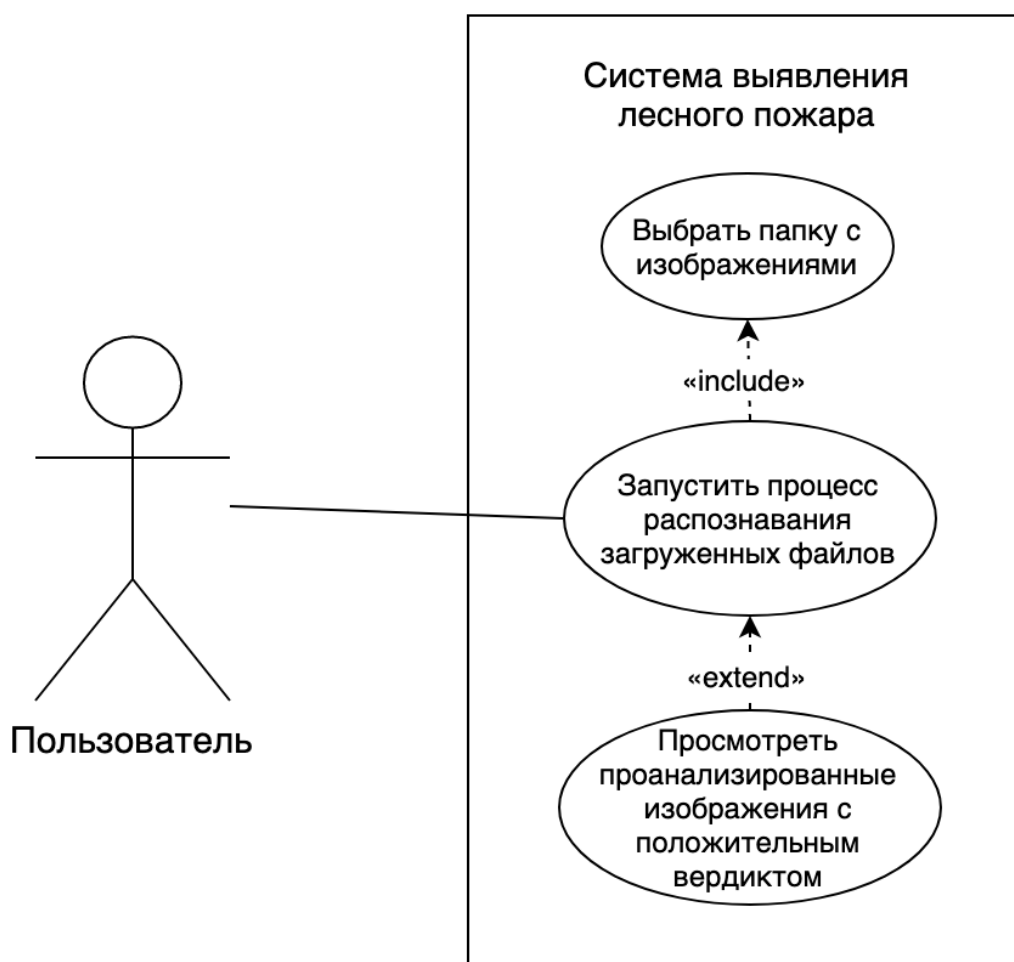


Рис. 8. Диаграмма вариантов использования

разработана диаграмма деятельности (рис. 9). На данной диаграмме изображен процесс взаимодействия пользователя с системой выявления лесного пожара.

Выводы по третьей главе

В третьей главе, в результате рассмотренной литературы и уже существующих аналогов и решений, были определены функциональные и нефункциональные требования и представлены диаграммы деятельности и вариантов использования. Спроектирована система распознавания пожара на изображении, состоящая из клиентского приложения и серверной части, реализация которой представлена в четвертой главе.

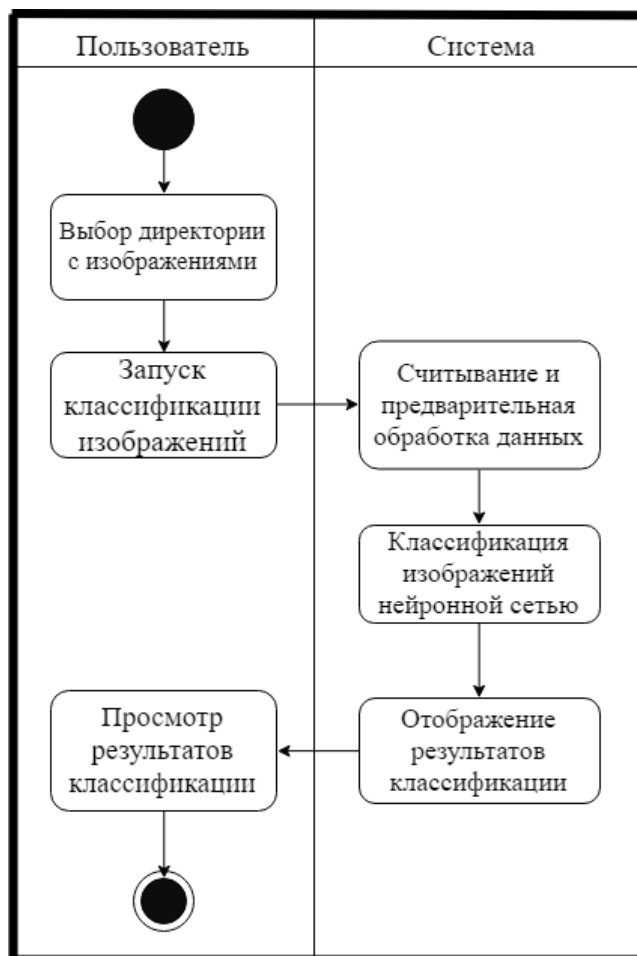


Рис. 9. Диаграмма деятельности

4. РЕАЛИЗАЦИЯ

4.1. Программные средства реализации

Для разработки программной части для распознавания пожара на изображении был использован высокоуровневый язык программирования Python 3.7. Разработка системы велась в среде разработки PyCharm 2018.2.3 (Community Edition) [6] и на операционной системе Windows 7 [8]. Для создания нейронной сети были использованы следующие библиотеки: cv2 (OpenCV) – для работы с изображениями, Keras – для работы (и в том числе для создания) с нейронными сетями, TensorFlow – база для создания и использования нейронных сетей, Numpy [9] – для работы с большими массивами и матрицами данных и Collection [1] – для работы со списками. Для написания интерфейса была использована библиотека PyQt5 [12]. Для написания серверной части использовался веб-фреймворк для Python (REST API) – Flask [4], а также Keras, cv2 и Numpy. Для написания клиентской части приложения использовались: Requests [13] – для создания запросов к серверу (REST API), Sys [14] – системная библиотека, cv2 и Numpy.

Формирование обучающей выборки

Поскольку данные для обучения и тестирования специфичны, их нет в открытом доступе. Поэтому для обучения и тестирования нейронной сети была использована база изображений Корсиканского университета во Франции (Corsican Fire Database), включающая в себя изображения в тепловом инфракрасном фильтре [2]. Для обучения нейронной сети было использовано 702 изображения с пожаром. Изображения без пожара (961 фотография) были взяты с видео-хостинга YouTube [18] путем фрейминга 21 видеозаписи. Общее объем обучающей выборки составил 1663 изображения (88 % от всей выборки).

Общий объем тестовой выборки составил 210 изображение, из которых 88 с пожаром и 122 без (12 % от всей выборки). Примеры входных данных изображены на рис. 10 и 11.



Рис. 10. Пример входных данных с пожаром



Рис. 11. Пример входных данных без пожара

4.2. Реализация нейронной сети

На входной слой сети подается черно-белое изображение, размером 128x72 пикселей. На выходе получаем вердикт двоичного типа, классифицирующий входное изображение либо как содержащее пожар (1), либо не содержащее (2). Реализация нейронной сети основана на топологии из главы Проектирование. В качестве функции ошибки была выбрана «categorical_crossentropy». Для реализации нейронной сети использовалась библиотека Keras.

4.3. Реализация серверной части системы

Для реализации серверной части использовалась библиотека flask. Данная библиотека поддерживает протокол передачи сообщений REST API. Из четырех методов, поддерживаемых REST API (POST, GET, PUT, DELETE) нам необходим только один – POST. С помощью этого метода мы можем послать в качестве запроса файл (изображение) и получить в качестве ответа на запрос вердикт о том, присутствует ли на данном изображении огонь. Протокол клиент-серверного взаимодействия представлен на рис. 13.

Перед вынесением вердикта, система проводит предобработку полученных данных. Программный код предобработки тоже хранится на сервере и выполняется непосредственно перед вынесением вердикта. Программный код предобработки изображения представлен на рис. 12.

```
def preprocess(image):
    image = cv2.resize(image, (128, 72))
    if len(image.shape) == 3 and image.shape[-1] == 3:
        image = cv2.cvtColor(image, cv2.COLOR_RGB2GRAY)
        image = np.expand_dims(image, -1)
    elif len(image.shape) == 3 and image.shape[-1] == 4:
        image = cv2.cvtColor(image, cv2.COLOR_RGBA2GRAY)
        image = np.expand_dims(image, -1)
    if len(image.shape) == 2:
        image = np.expand_dims(image, -1)
    return image
```

Рис. 12. Программный код предобработки изображения

На серверной части используются вспомогательная функция. Задача вспомогательной функции заключается обработке изображения с использованием обученной модели нейронной сети с целью вынесения вердикта.

Расположение программного кода серверной части и обученной модели нейронной сети находится на серверах облачной платформы Amazon Web Services (EC2). Этот взб-хостинг был выбран из-за наличия тестового бесплатного периода обслуживания.

```
//Отправка запроса с файлом на сервер
{
  "files": {
    "file": "<бинарное представление файла>"
  }
}

//Получение результата запроса с сервера
{
  "filename": "<filename:string>",
  "verdict": <verdict:int>
}
```

Рис. 13. Протокол клиент-серверного взаимодействия

4.4. Реализация клиентской части системы

Клиентская часть состоит из интерфейса и алгоритма отправки изображения и получения вердикта с предоставлением результатов пользователю.

Интерфейс реализован в соответствии с проектированием из прошлой главы. Для его реализации была использована библиотека PyQt5. Функционал данной библиотеки полностью удовлетворил все требования к визуальному оформлению клиентской части. Пример пользовательского интерфейса представлен на рис. 14.

Алгоритм отправки изображения и получения вердикта с предоставлением результатов пользователю был реализован с использованием двух вспомогательных функций. Первая функция получает на вход путь до директории и считывает из нее все файлы с разрешением

изображений (.jpg, .png). Для этого используется функция `getExistingDirectory()` класса `QFileDialog` библиотеки `PyQt5`. При вызове данная функция позволяет выбрать директорию в открывшемся проводнике и возвращает в качестве результата абсолютный путь до выбранной директории. Далее при помощи библиотеки `os` мы получаем полные имена всех файлов, хранящихся в выбранной папке и проверяем их расширения. Если расширение файла соответствует требованию (.jpg, .png), то данный файл считывается с помощью функции `imread()` библиотеки `OpenCV`.

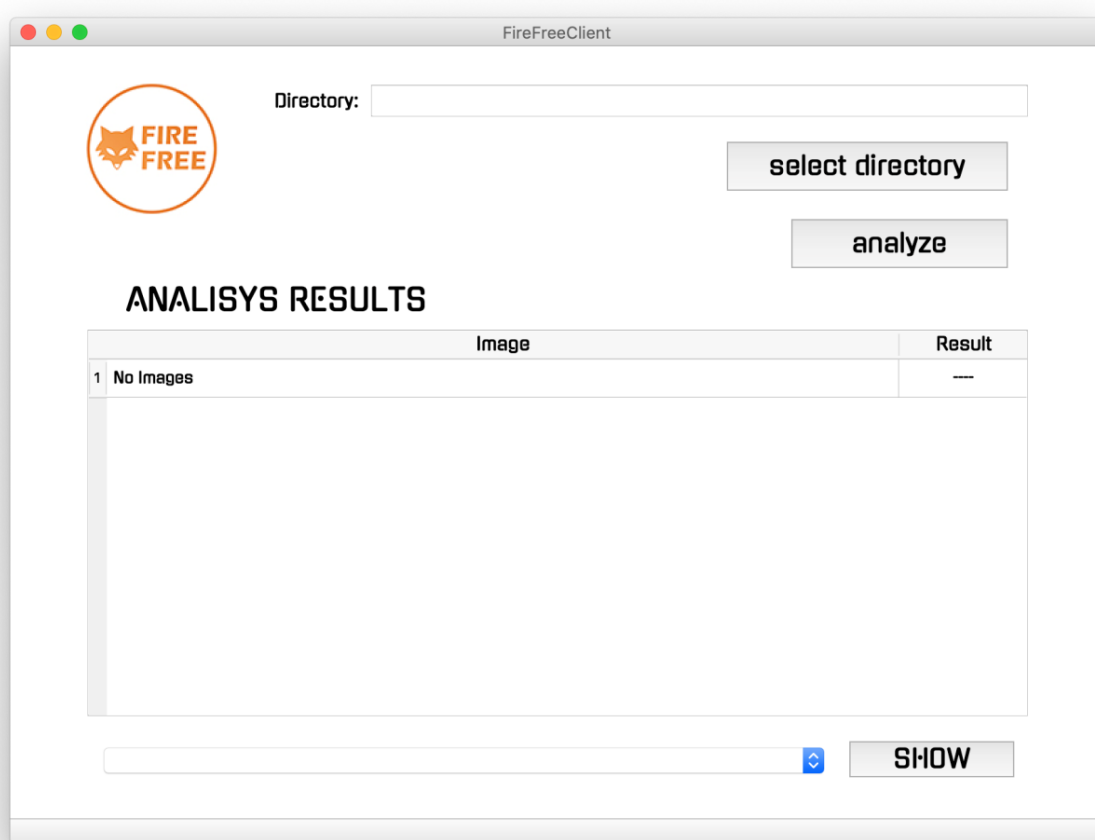


Рис. 14. Пользовательский интерфейс

Вторая функция отвечает за последовательную передачу, считанных из выбранной пользователем директории, изображений на серверную часть с целью дальнейшего анализа с использованием нейронной сети, получение результата от сервера и предоставление результатов пользователю в удобном

для восприятия виде. Для отправки на сервер используется функция `post()` библиотеки `requests`. В качестве ответа сервера мы получаем имя файла и результат аналитической обработки данного файла в виде номера класса, к которому модель нейронной сети присвоила изображение. Далее пользователю предоставляется возможность посмотреть изображения, классифицированные как изображение с пожаром, в отдельном окне. Для отображения изображений в отдельном окне используется функция `imshow()` библиотеки `OpenCV`.

Выводы по четвертой главе

В данной главе были описаны средства разработки системы, а также реализация нейронной сети, пользовательского интерфейса и веб сервера. Разработанная система соответствует всем, описанным в третьей главе, требованиям. Описаны обучающая и тестовые выборки. Представлен графический интерфейс приложения.

5. ТЕСТИРОВАНИЕ

5.1. Результаты тестирования

5.1.1. Тестирование нейронной сети

В ходе выполнения работы была составлена обучающая и контрольная выборки. Нейронная сеть была обучена на 124 эпохах. При обучении сеть показала значение точности в 97 %, а функция потерь составила 0,092. Обучение проводилось на компьютере с видеокартой nVidia GeForce GTX 1050 и оперативной памятью 8 ГБ.

5.1.2. Функциональное тестирование

Функциональное тестирование – тестирование, направленное на проверку корректности работы функциональности приложения (корректность реализации функциональных требований) [19].

Тест № 1. Цель: Проверить возможность выбора директории.

Действие: Пользователь нажимает на кнопку «select directory», выбирает директорию и нажимает кнопку «open».

Ожидаемый результат: Пользователь увидит окно выбора директории, при выборе директории и последующем нажатии на кнопку «open» пользователь увидит абсолютный путь до выбранной директории в специально отведенном для этого месте в интерфейсе.

Тест пройден? Да.

Тест № 2. Цель: Проверить возможность анализа изображений из выбранной директории.

Действие: Пользователь нажимает на кнопку «Analyze».

Ожидаемый результат: Таблица результатов будет заполнена в соответствии с изображениями в выбранной директории и результатом работы серверной части программы. В блоке селекторе появятся наименования изображений с обнаруженным на них очагом возгорания.

Тест пройден? Да.

Тест № 3. Цель: Проверить возможность просмотра изображений, классифицированных как изображения с пожаром, в отдельном окне.

Действие: Пользователь выбирает в селекторе наименования изображения, которое он желает просмотреть и нажимает на кнопку «Show».

Ожидаемый результат: Появляется новое окно с изображением, соответствующим выбранному в селекторе наименованию.

Тест пройден? Да.

Тест № 4. Цель: Повторно выбрать директорию для анализа.

Действие: Выбрав директорию и проанализировав изображения в ней пользователь еще раз нажимает на кнопку «select directory», выбирает новую директорию и нажимает кнопку «Open».

Ожидаемый результат: Пользователь увидит окно выбора директории, при выборе директории и последующем нажатии на кнопку «Open» пользователь увидит абсолютный путь до новой выбранной директории в специально отведенном для этого месте в интерфейсе.

Тест пройден? Да.

Тест №5. Цель: Повторно проанализировать изображения из повторно выбранной директории.

Действие: Пользователь нажимает на кнопку «analyze».

Ожидаемый результат: Таблица результатов будет заполнена в соответствии с изображениями в перевыбранной директории и результатом работы серверной части программы. В блоке селекторе появятся наименования изображений с обнаруженным на них очагом возгорания. Данные о предыдущих аналитических процессах будут удалены. Их места займут новые данные.

Тест пройден? Да.

Выводы по пятой главе

В ходе тестирования были подготовлены тесты для функционального тестирования, проведено обучение нейронной сети и тестирование ее работы

(на подготовленной тестовой выборке). Была протестирована работа единой системы распознавания лесного пожара. Система выполнила свою задачу и осуществила классифицирование всего набора входных изображений.

ЗАКЛЮЧЕНИЕ

В соответствии с целями данной работы был проведен обзор научной литературы и существующих аналогов, который подтвердил актуальность поставленной задачи. В ходе анализа были сформулированы следующие подзадачи: выделить пятно повышенной температуры на изображении и проанализировать полученную информацию с помощью сверточной нейронной сети.

Была рассмотрена теоретическая часть сверточных нейронных сетей (принципы их работы, информация о слоях свертки и подвыборки). Был сформулирован процесс предобработки изображения.

Были определены функциональные и нефункциональные требования и представлены диаграммы деятельности и вариантов использования, а также спроектирована система распознавания пожара на изображении, состоящая из клиентского приложения и серверной части.

Представлена реализация нейронной сети, пользовательского интерфейса и веб сервера. Разработанная система соответствует всем требованиям. Описаны обучающая и тестовая выборки.

Была протестирована работа нейронной сети и единой системы распознавания. Система выполнила свою задачу и осуществила классифицирование набора входных изображений.

В рамках дипломного проекта было разработано программное обеспечение для выявления лесного пожара, состоящее из клиентского приложения и серверной части.

Были решены следующие задачи.

1. Был проведен анализ программных аналогов и научной литературы в предметной области.
2. Была подготовлена обучающую и тестовую выборки.
3. Разработана нейронная сеть для распознавания пожара.
4. Обучена и протестирована нейронная сеть.

5. Реализована система выявления лесного пожара, состоящая из клиентского приложения и серверной части.

6. Протестирована работа системы.

ЛИТЕРАТУРА

1. Collection. Документация. [Электронный ресурс] <https://docs.python.org/2/library/collections.html> (дата обращения: 01.04.2019).
2. Corsican Fire forest Database. Официальный сайт. [Электронный ресурс] URL: <http://cfdb.univ-corse.fr/> (дата обращения: 25.03.2019).
3. FireWatch. Официальный сайт. [Электронный ресурс] URL: <http://www.iq-firewatch.com/> (дата обращения: 04.02.2019).
4. Flask. Официальный сайт. [Электронный ресурс] URL: <http://flask.pocoo.org/> (дата обращения: 07.03.2019).
5. Horng W. -B., Peng J.-W. Image based fire detection using neural networks.// ResearchGate, 2006.
6. JetBrains. Официальный сайт. [Электронный ресурс] URL: <https://www.jetbrains.com/pycharm> (дата обращения: 17.03.2019).
7. Keras. Официальная документация. [Электронный ресурс] URL: <http://keras.io> (дата обращения: 21.04.2018).
8. Microsoft. Официальный сайт. [Электронный ресурс] URL: <https://www.microsoft.com> (дата обращения: 17.03.2019).
9. Numpy. Официальная документация. [Электронный ресурс] URL: <http://www.numpy.org/> (дата обращения: 12.03.2019).
10. Object recognition with gradient-based learning, Y. LeCun, P. Haffner, L. Botton, Y. Bengio. AT&T Shannon Lab – 24 с.
11. OpenCV. Официальная документация. [Электронный ресурс] URL: <http://opencv.org> (дата обращения: 21.04.2018).
12. PyQt5. Официальный сайт. [Электронный ресурс] URL: <https://www.riverbankcomputing.com/static/Docs/PyQt5> (дата обращения: 12.03.2019).
13. Requests. Официальная документация. [Электронный ресурс] URL: <https://2.python-requests.org> (дата обращения 12.03.2019).
14. Sys. Официальная документация. [Электронный ресурс] URL: <https://docs.python.org/3.4/library/sys> (дата обращения: 12.03.2019).

15. TensorFlow. Официальная документация. [Электронный ресурс] URL: <https://www.tensorflow.org/> (дата обращения: 21.04.2018).
16. Yansong Y. G., Liu Y. C., Liu Y., Forest Fire Detection Using Artificial Neural Network Algorithm Implemented in Wireless Sensor Networks. ZTE Communications, 2015.
17. Yoshua Bengio A.C., Goodfellow Ian. Deep Learning. – MIT Press, 2016.
18. YouTube. Официальный сайт. [Электронный ресурс] URL: <https://www.youtube.com> (дата обращения: 02.02.2019).
19. Брамсон М.А. Инфракрасное излучение нагретых. – Издательство Наука, 1964 – 226 с.
20. Куликов С.С. Тестирование программного обеспечения. Базовый курс. 2 издание. – Издательство Четыре Четверти, 2017.
21. Классификация. [Электронный ресурс] URL: <http://www.machinelearning.ru/wiki/index/php&title=Классификация> (дата обращения: 02.02.2019).
22. МВД. Официальный сайт. [Электронный ресурс] URL: <http://39.xn--b1aew.xn--p1ai/news/item/1138231> (дата обращения: 17.04.2018).
23. Новости MailRu. [Электронный ресурс] URL: <https://news.mail.ru/incident/33191487/?frommail=1> (дата обращения: 17.03.2018).
24. МЧС. Официальный сайт. [Электронный ресурс] URL: <http://39.mchs.gov.ru/pressroom/news/item/1292150> (дата обращения: 17.04.2018).
25. Нейронные сети. Часть 1. [Электронный ресурс] URL: <https://habr.com/ru/post/312450/> (дата обращения: 17.03.2018).
26. Причины пожаров в лесу. [Электронный ресурс] URL: <https://protivpozhara.com/tipologija/prirodnye/vidy-lesnyx-pozharov> (дата обращения: 02.02.2019).

27. Р. Д. Б. М., UML 2.0. Объектно-ориентированное моделирование и разработка. Питер, 2-е издание, 2007 – 544 с.

28. Романов В. П. Интеллектуальные информационные системы в экономике. Учебное пособие. – Издательство ЭКЗАМЕН, 2003 – 496 с.

29. Статья Tensorflow. [Электронный ресурс] URL:
<https://ru.wikipedia.org/wiki/TensorFlow> (дата обращения: 05.03.2019).

30. Статья Keras. [Электронный ресурс]
URL:<https://ru.wikipedia.org/wiki/Keras> (дата обращения: 05.03.2019).

31. Статья OpenCV. [Электронный ресурс] URL:
<https://ru.wikipedia.org/wiki/OpenCV> (дата обращения: 05.03.2019).

32. Хайкин С. Нейронные сети. Полный курс. / Под ред. Н.Н. Куссуль
Второе издание. – ООО Вильямс, 2016 – 1104 с.

ПРИЛОЖЕНИЯ

Приложение 1

Скриншоты работы системы

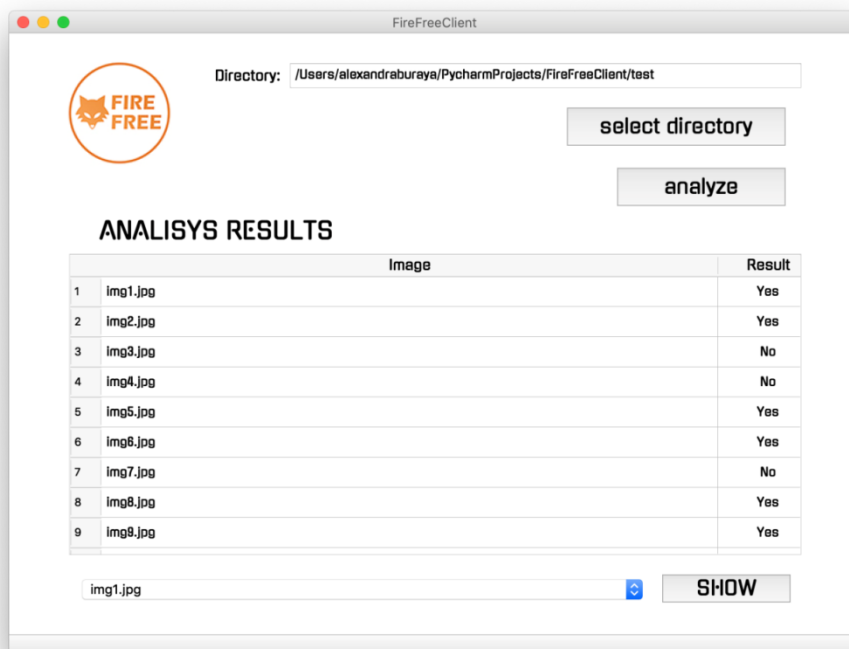


Рис. 1. Результат анализа изображений

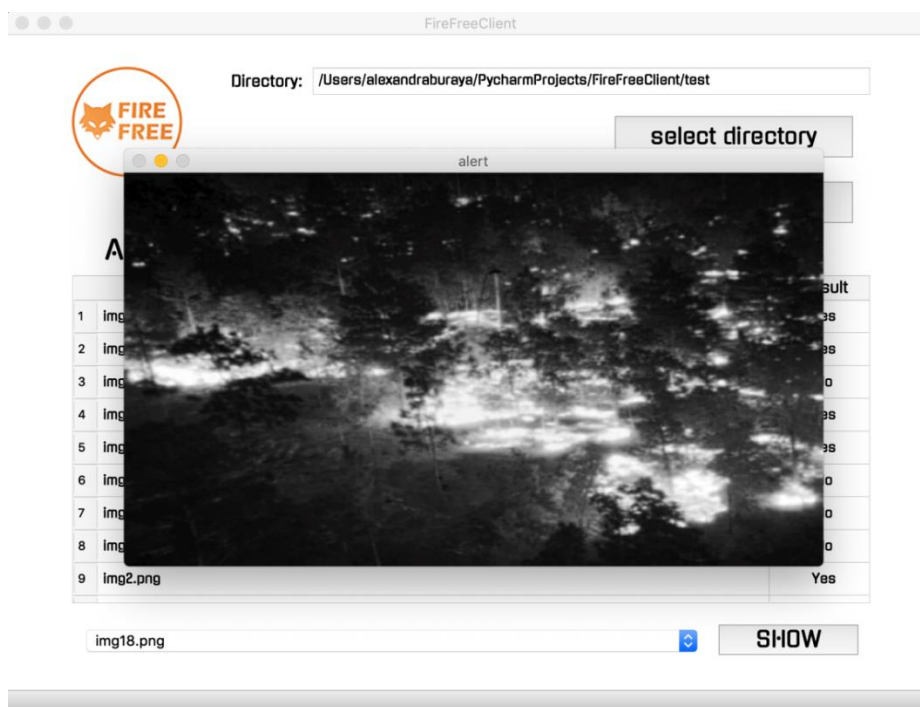


Рис. 2. Просмотр изображения с идентифицированным пожаром

Приложение 2

Листинг программного кода

```
@app.route('/', methods=['POST'])
def index():
    response = {}
    if 'file' not in request.files:
        response['response'] = 'No file provided'
    else:
        response['response'] = request.files['file'].filename
        image = preprocess(request.files)
        image_mass = [image]
        image_mass = np.array(image_mass)
        predictions = model.predict(image_mass)
        verdict = predictions.argmax()
        response['verdict'] = verdict
    return jsonify({
        'response': response,
        'request': request.form
    })
```

Рис. 3. Листинг программного кода серверной обработки сообщения, полученного с клиента и отправки ответа

```
def preprocess(image):
    image = cv2.resize(image, (128, 72))
    if len(image.shape) == 3 and image.shape[-1] == 3:
        image = cv2.cvtColor(image, cv2.COLOR_RGB2GRAY)
        image = np.expand_dims(image, -1)
    elif len(image.shape) == 3 and image.shape[-1] == 4:
        image = cv2.cvtColor(image, cv2.COLOR_RGBA2GRAY)
        image = np.expand_dims(image, -1)
    if len(image.shape) == 2:
        image = np.expand_dims(image, -1)
    return image
```

Рис. 4. Листинг программного кода предобработки изображения