МИНОБРНАУКИ РОССИИ
Федеральное государственное бюджетное образовательное учреждение
высшего образования
«МИРЭА – Российский технологический университет»
РТУ МИРЭА

ОТЧЁТ ПО ПРАКТИКЕ В ПРОФИЛЬНОЙ СФЕРЕ

Тема практики: «Подготовка и предварительная обработка данных предметной области для модели распознавания вариант 510»

Отчет представлен к рассмотрению:
Студент группы ИКБО-11-23 «31» мая 2025 г. _____ Мусатов И.А.
(подпись)

Отчет утвержден.
Допущен к защите:

Руководитель практики «31» мая 2025 г. _____ Магомедов Ш.Г.
(подпись)

Москва 2025 г.

МИНОБРНАУКИ РОССИИ
Федеральное государственное бюджетное образовательное учреждение
высшего образования
«МИРЭА – Российский технологический университет»
РТУ МИРЭА

ИНДИВИДУАЛЬНОЕ ЗАДАНИЕ НА ПРАКТИКУ В ПРОФИЛЬНОЙ СФЕРЕ

Студенту 2 курса учебной группы ИКБО-11-23
Мусатову Ивану Алексеевичу

Время практики: с 26 мая 2025 г. по 31 мая 2024 г.
Должность на практике: студент

1. ЦЕЛЕВАЯ УСТАНОВКА: подготовка и предварительная обработка данных предметной области, а также формулировка аналитической задачи, решаемой с использованием технологий искусственного интеллекта

2. СОДЕРЖАНИЕ ПРАКТИКИ:

2.1 Обзор существующих решений и подходов к подготовке данных и построению моделей анализа в выбранной предметной области

2.2 Описание источников, структуры, характеристик и объема исходных данных, включая потенциальные проблемы качества данных и способы их устранения

2.3 Формулировка задачи анализа данных с указанием её целей, постановки в терминах машинного обучения, а также описание ожидаемого результата

3. ОРГАНИЗАЦИОННО-МЕТОДИЧЕСКИЕ УКАЗАНИЯ: в процессе практики рекомендуется использовать периодические издания и отраслевую литературу годом издания не старше 10 лет.

СОГЛАСОВАНО:
Руководитель практики
«26» мая 2025 г. _____ (Магомедов Ш.Г.)
(подпись)

Задание получил
«26» мая 2025 г. _____ (Мусатов И.А.)
(подпись)

МИНОБРНАУКИ РОССИИ
Федеральное государственное бюджетное образовательное учреждение
высшего образования
«МИРЭА – Российский технологический университет»
РТУ МИРЭА

РАБОЧИЙ ГРАФИК ПРОВЕДЕНИЯ
практики в профильной сфере

студента Мусатова И.А. 2 курса группы ИКБО-11-23.

№	Сроки выполнения	Этап	Отметка о выполнении
1	26.05.2025	Подготовительный этап, включающий в себя организационное собрание (Вводная лекция о порядке организации и прохождения практики)	выполнено
2	с 26.05.2025 по 29.05.2025	Выполнение задания по практике в соответствии с выданным заданием студента. (Мероприятия по сбору, обработке и структурированию материала, выполнение поставленной задачи)	выполнено
3	с 29.05.2025 по 30.05.2025	Подготовка отчета по практике (Оформление материалов отчета в полном соответствии с требованиями на оформление учебных работ студентов)	выполнено
4	31.05.2025	Представление отчета по практике к защите посредством загрузки на портал ДПО (https://online-dpo.mirea.ru/)	выполнено

Согласовано:

Руководитель практики _____/ Магомедов Ш.Г. /

Обучающийся _____/ Мусатов И.А. /

Фактическая тема практики: «Подготовка и предварительная обработка данных предметной области для модели детекции возгорания на изображении»

СОДЕРЖАНИЕ

ВВЕДЕНИЕ 6
1 ОСНОВНАЯ ЧАСТЬ 8
1.1 Анализ предметной области 8
1.2 Обзор инструментов и решений для подготовки данных 9
1.3 Поиск и выбор данных 10
1.4 Характеристика и статистический анализ данных 11
1.5 Выявление потенциальных проблем 18
1.6 Решение проблем 19
1.7 Постановка задачи анализа данных 21
1.8 Предполагаемое архитектурное решение 22
ЗАКЛЮЧЕНИЕ 23
СПИСОК ИНФОРМАЦИОННЫХ ИСТОЧНИКОВ 25

ВВЕДЕНИЕ

Несмотря на активное развитие и внедрение противопожарных технологий, пожары остаются одной из существенных угроз безопасности людей и инфраструктуры. Согласно статистике [1], показатель смертности граждан в результате пожаров на территории России выше, чем в других развитых странах. А размеры экономического ущерба оцениваются в миллиардах рублей.

В связи с этим актуальной задачей является улучшение и модернизация противопожарных технологий. Наиболее востребованными являются превентивные меры, либо же методы, позволяющие обнаруживать возгорания на их ранней стадии. Стандартные решения, будь то датчики дыма или тепловые сенсоры, применимы лишь на ограниченных пространствах. Наиболее надежным решением (впрочем, как и в любой области, где требуется оценка ситуации) является человеческий мониторинг. Очевидно, что данный подход очень требователен и не может быть использован массово. В таких случаях на помощь приходит автоматизация. В частности, делегирование наблюдения и оценки системам видеонаблюдения с алгоритмами компьютерного зрения.

Подобные интеллектуальные алгоритмы основываются на процессе обучения на данных, т.е. накоплении и интерпретации определенных признаков, которые в будущем позволят алгоритму обобщать свою «оценку» на произвольные входные данные.

То, какие признаки будет накапливать модель, целиком определяется обучающим набором данных. Поэтому процесс сбора, подготовки и предобработки данных является ключевым для последующего формирования качественной и способной к обобщению интеллектуальной модели.

Таким образом, целью данной практики является подготовка и анализ набора данных для обучения модели детекции возгорания на изображении.

В процессе предварительной обработки набора данных будут решаться следующие задачи:

- подробный анализ предметной области с выявлением потенциальных аномалий;
- обзор существующего инструментария подготовки данных;
- поиск и выбор подходящего набора данных;
- описание характеристик набора данных с указанием проблем;

По результатам предобработки данных будет сформулирована конкретная задача анализа данных для достижения ожидаемых результатов в предметной области.

1 ОСНОВНАЯ ЧАСТЬ

1.1 Анализ предметной области

Тема распознавания пожаров и возгораний сопряжена с большим количеством потенциальных аномалий. Даже при ручном (человеческом) мониторинге не исключены ошибки неправильного восприятия объектов.

Огонь можно охарактеризовать, в первую очередь, как яркое световое явление с динамичной природой. Тем не менее, в природе (как естественной, так и искусственной) существует немало объектов с похожей структурой. Например, движущиеся источники света (скажем, фары), мерцающие лампы и фонари, недаром называющиеся «огнями». Блики на водной глади, металлах и стекле также при определенных условиях могут быть восприняты за возгорание.

Дым – верное следствие возгорания – также не уникален по своей природе. Его рассеянность и сероватый оттенок могут быть спутаны с туманом или выхлопными газами. Тем не менее, динамичность дыма позволяет отделять его от остальных привычных газообразных структур.

Но, пожалуй, более существенной и трудно устранимой проблемой является оценка вредоносности огня. Имеется в виду отличие огня в камине или мангале от горящего кресла рядом с камином или горящего куста рядом с мангалом. Человек-наблюдатель в большинстве случаев легко может определить, несет данный огонь пользу или вред. Но как это можно формализовать? Какими признаками можно разделить две ситуации? Возможно, различие как раз и достигается за счет окружающего контекста. Как в рассмотренных примерах – кресла и камина. Либо же подсказкой может стать динамика увеличения возгорания. В случае с камином огонь сохраняет свое положение. В случае же с возгоранием, область поражения будет распространяться, тем самым и обеспечивая бесконтрольность ситуации.

Проведенное рассмотрение позволяет нам заключить, что для создания обобщающей и корректной модели необходимо включить в набор данных помимо примеров изображений с возгоранием и дымом также и «аномальные» изображения с источниками света, бликами, туманом и «контролируемым» огнем.

1.2 Обзор инструментов и решений для подготовки данных

Чтобы зафиксировать изображение возгорания (неконтролируемого огня), естественным образом требуется наличие самой неконтролируемой ситуации. Подобные мероприятия в экспериментальных условиях представляют значительные риски. А в реальных условиях, мягко говоря, фиксация изображений имеет далеко не первостепенное значение. Поэтому было бы неплохо иметь методы, позволяющие безопасно имитировать неконтролируемые возгорания.

Такие методы уже существуют и активно используются. В частности, существует метод синтетической генерации. Например, в работе [2] описывается настройка пайплайна (pipeline) по генерации синтетических данных для обучения моделей детекции пожаров с помощью цифровых двойников (digital twins). Метод представляет собой создание физически корректных моделей огня (в работе для этого используется движок Unity3D) и внедрение их в реальные сцены с видеокамер. Использование симуляций, как показывает исследование, не ухудшает показатели моделей на тестовой выборке, предоставляя в среднем точность 0,90 (mean average precision).

Помимо генерации принципиально новых изображений существует подход, позволяющий выделять качественно новые данные из уже существующего набора. Это метод аугментации (augmentation). По сути, это набор методов физического изменения и искажения данных. Например, поворот, отражение, масштабирование, размытие, изменение яркости и прочих фильтров.

Одной из самых многофункциональных и удобных библиотек для осуществления аугментации данных является библиотека Albumentations [3]. Помимо прочего она позволяет осуществлять сложные преобразования даже bbox-данных, т.е. данных с размеченными рамками (bounding boxes). В контексте обучения модели детекции возгорания это особенно полезная функциональность.

1.3 Поиск и выбор данных

Пожалуй, самым времязатратным и ответственным этапом в подготовке набора данных является поиск подходящего датасета (если, конечно, решение о самостоятельном сборе было отвергнуто).

Существует некоторое количество платформ и ресурсов, предоставляющих возможность поиска наборов данных. Воспользуемся наиболее популярными из них:

- Kaggle (платформа, организующая соревнования по ML и сбор датасетов);
- Roboflow (платформа для создания и развертывания моделей компьютерного зрения);
- Dataset Search (поисковый сервис для открытых наборов данных).

В результате поиска наборов данных по тегам fire, smoke, detection были выделены несколько релевантных экземпляров. Опишем их преимущества и недостатки подробнее.

Несмотря на большое разнообразие, датасеты с платформы Kaggle имеют различные недостатки. Тезисно опишем их:

- Smoke Detection Dataset – набор данных содержит изображения дыма, но не огня;

- Smoke-Fire-Detection-YOLO – объемный набор (40 тысяч изображений), но специализирующийся на обширных изображениях, сделанных с видеокамер вышек наблюдения;

- Fire and Smoke Dataset – маленький объем данных (200 изображений в открытом доступе). Для полноценного обучения не хватит;

- Forest_Fire_Smoke_and_Non_Fire_Image_Dataset – специализация на лесных пожарах. Локальные возгорания не предоставлены;

Поиск посредством Dataset Research приводил либо к лицензионным наборам данных, либо к наборам, объема которых для обучения недостаточно. Тем не менее, были даны ссылки на датасеты с платформы Roboflow.

Поиск данных на платформе Roboflow потребовал регистрации, но зато привел к продуктивным результатам. Были выделены два релевантных набора:

- Fire-Smoke-Detection-Yolov11 – набор с разнообразием локаций возгорания. Но отсутствует деление на огонь и дым. Объем – 4 тысячи изображений;

- FireSmokeDataset – также содержит разнообразные локации возгорания. Имеются «аномальные» данные, ведется классификация bbox по трем классам «огня», «дыма» и «прочего». Объем – 20 тысяч изображений;

В результате разбора недостатков всех вышеназванных наборов было принято решение об использовании датасета FireSmokeDataset с проведением дополнительной обработки.

1.4 Характеристика и статистический анализ данных

Приведем характеристику данных выбранного датасета.

Набор данных представляет собой директорию с тремя разделенными выборками: train, test и validation. В каждой из папок с выборкой располагаются изображения в формате JPG, а также аннотационный файл.

Аннотации оформлены в нотации COCO (common objects in context). Такой формат предполагает структурирование файла на пять частей: описание датасета, список изображений (названий файлов), список категорий (классов), аннотации для задач обнаружения, аннотации для задач сегментации (в данной работе применяться не будет) [4]. Структура позволяет организовывать данные для многоклассовой детекции. Из особенностей формата стоит также отметить способ указания рамок объектов (bounding box). Они производятся в форме "bbox": [x, y, width, height], где x и y – минимальные координаты прямоугольника по осям x и y соответственно. Такая структура bbox используется не во всех форматах, поэтому будем учитывать эту особенность.

На рисунке 1.1 приведено изображение внешнего вида файла аннотаций.

Рисунок 1.1 – Содержимое аннотационного файла

Проанализируем теперь содержание датасета в количественной форме. Для этого напишем небольшой скрипт, собирающий статистику по данным. Программная реализация скрипта представлена на рис. 1.2.

```python
import json
from collections import Counter, defaultdict

def load_coco_json(json_path):
    with open(json_path, 'r', encoding='utf-8') as f:
        return json.load(f)

def get_category_info(coco_data):
    return {cat["id"]: cat["name"] for cat in coco_data["categories"]}

def analyze_annotations(coco_data):
    annotations = coco_data["annotations"]
    images = coco_data["images"]
    category_names = get_category_info(coco_data)
    category_counter = Counter()
    for ann in annotations:
        category_counter[ann["category_id"]] += 1
    category_counter_named = {category_names[k]: v for k, v in category_counter.items()}
    return {
        "num_images": len(images),
        "num_annotations": len(annotations),
        "category_counts": category_counter_named
    }

def print_report(stats, sample):
    print(f"\nОбщая статистика по {sample}:")
    print(f"Изображений: {stats['num_images']}")
    print(f"Аннотаций: {stats['num_annotations']}")
    print("Кол-во аннотаций по категориям:")
    for category, count in stats['category_counts'].items():
        print(f"  - {category}: {count}")

for sample in ['train', 'test', 'valid']:
    coco_data = load_coco_json(f'./Fire-smoke-dataset/{sample}/_annotations.coco.json')
    stats = analyze_annotations(coco_data)
    print_report(stats, sample)
```

[24] ✓ 0.2s

Рисунок 1.2 – Скрипт для подсчета объемов данных

Вывод алгоритма позволяет дать следующую количественную характеристику набору данных (см. таблицу 1).

Таблица 1 – Статистика объемов данных по выборкам

Характеристика	**train**	**test**	**validation**
Кол-во изображений	12813	2237	6068
Аннотаций Smoke	12348	2466	4869
Аннотаций Fire	14153	2161	5986
Аннотаций Other	3796	804	2135
Аннотаций всего	30297	5431	12990

Помимо объемов рассмотрим также и частоту сочетаемости различных классов друг с другом. Для этого напишем следующий код (см. рис. 1.3).

```python
import json
import matplotlib.pyplot as plt
from collections import defaultdict, Counter

with open("Fire-smoke-dataset/train/_annotations.coco.json", "r", encoding="utf-8") as f:
    coco = json.load(f)

images = {img["id"]: img for img in coco["images"]}
categories = {cat["id"]: cat["name"] for cat in coco["categories"]}
annotations = coco["annotations"]

image_to_anns = defaultdict(list)
class_counter = Counter()
combo_counter = Counter()
bbox_areas = []

for ann in annotations:
    image_to_anns[ann["image_id"]].append(ann)
    class_counter[ann["category_id"]] += 1
    bbox = ann["bbox"]
    area = bbox[2] * bbox[3]
    bbox_areas.append(area)

for img_id, anns in image_to_anns.items():
    class_combo = tuple(sorted(set(ann["category_id"] for ann in anns)))
    combo_counter[class_combo] += 1

plt.figure(figsize=(6, 6))
labels = [", ".join(categories[c] for c in k) for k in combo_counter.keys()]
sizes = list(combo_counter.values())
plt.pie(sizes, labels=labels, autopct="%1.1f%%", startangle=30)
plt.title("Сочетания классов на изображениях")
plt.axis("equal")
plt.show()

plt.figure(figsize=(6, 4))
plt.hist(bbox_areas, bins=40, log=True, color='skyblue')
plt.title("Площади bounding boxes")
plt.xlabel("Площадь (px²)")
plt.ylabel("Количество (логарифмическая шкала)")
plt.show()
```

[33] ✓ 0.3s

Рисунок 1.3 – Скрипт для выявления вспомогательных характеристик

В результате подсчетов можем построить следующую pie-chart диаграмму (см. рис. 1.4).

Рисунок 1.4 – Диаграмма частоты сочетания классов на изображении

На основе диаграммы сочетания классов можем сделать вывод, что основную часть набора данных составляют изображения, содержащие как возгорание, так и задымление, т.е. соответствуют предметной области. Вместе с тем также отметим, что изображений, на которых нет ни возгораний, ни задымлений крайне мало, что говорит о потенциальной проблеме нехватки «отрицательных» данных. Количественно процент «отрицательных» данных в датасете можно оценить как отношение числа аннотаций «other» к общему числу аннотаций. Что составляет 3796 / 30297 ≈ 0,125 = 12,5%. Рассмотрим этот момент подробнее в следующем пункте практики.

В скрипте, представленном на рис. 1.3 также проводился анализ размеров ограничивающих рамок. На основе подсчетов можем построить гистограмму распределения площадей рамок (представлена на рис. 1.5). Гистограмма показывает явное преобладание небольших рамок над рамками среднего и большого размера.

Чтобы оценить это количественно, воспользуемся вычислением медианы (см. рис. 1.6). Зная площадь изображений (а они являются

нормализованными по ширине и высоте – 640 на 640 пкс.), можем найти отношение медианного значения к ней. Получилось отношение, примерно равное 6%, что свидетельствует о преобладании рамок небольшого размера, и в перспективе позволит модели различать даже небольшие по размеру возгорания.

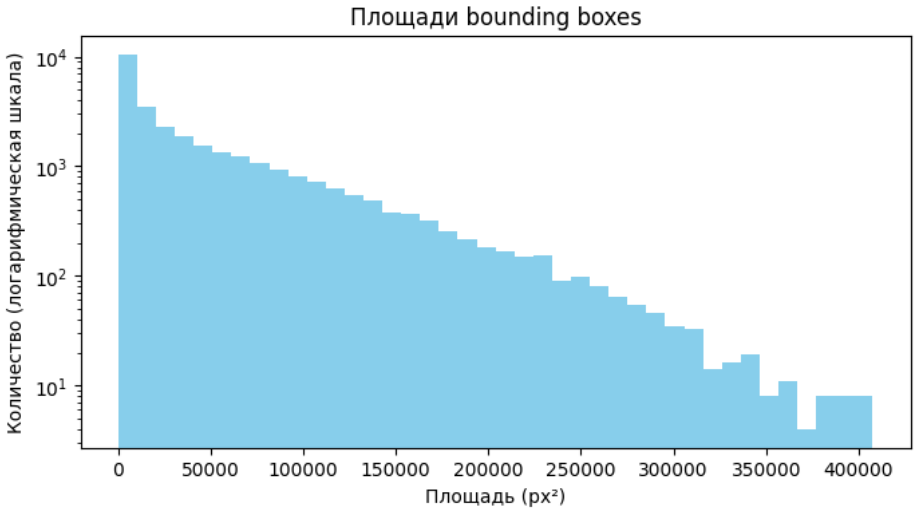

Рисунок 1.5 – Гистограмма распределения размеров bbox

```
import statistics

median_area = statistics.median(bbox_areas)
image_area = 640 * 640
print(f"Медианная площадь bbox: {median_area}")
print(f"Медианное отношение площадей: {median_area / image_area}")
```

```
[40]  ✓ 0.5s

...  Медианная площадь bbox: 25171.5
     Медианное отношение площадей: 0.061453857421875
```

Рисунок 1.6 – Медианные характеристики размеров bbox

Наконец, посмотрим на реализацию разметки изображений. Напишем вспомогательный скрипт, который будет визуализировать bbox на исходных изображениях (см. рис. 1.7).

На примере изображения сможем убедиться, в частности, в том, что в датасете учитываются и «аномалии». В данном случае, распознавание фонарей как класса «Other» (см. рис. 1.8).

```python
import cv2
import matplotlib.pyplot as plt

def draw_boxes_from_coco(image_id, images_dict, annotations_dict, category_id_to_name, image_root="."):
    image_info = images_dict[image_id]
    file_path = image_root + '/' + image_info["file_name"]
    image = cv2.imread(file_path)
    image = cv2.cvtColor(image, cv2.COLOR_BGR2RGB)

    for ann in annotations_dict.get(image_id, []):
        x, y, w, h = map(int, ann["bbox"])
        cat_id = ann["category_id"]
        class_name = category_id_to_name.get(cat_id, str(cat_id))
        cv2.rectangle(image, (x, y), (x + w, y + h), (0, 255, 0), 2)
        cv2.putText(image, class_name, (x, y - 5), cv2.FONT_HERSHEY_SIMPLEX, 0.6, (0, 255, 0), 2)

    plt.figure(figsize=(10, 8))
    plt.imshow(image)
    plt.axis("off")
    plt.show()


draw_boxes_from_coco(
    image_id=42,
    images_dict=images,
    annotations_dict=image_to_anns,
    category_id_to_name={1: "fire", 2: "other", 3: "smoke"},
    image_root=IMAGES_DIR
)
```

[17] ✓ 0.2s

Рисунок 1.7 – Скрипт для визуализации bbox

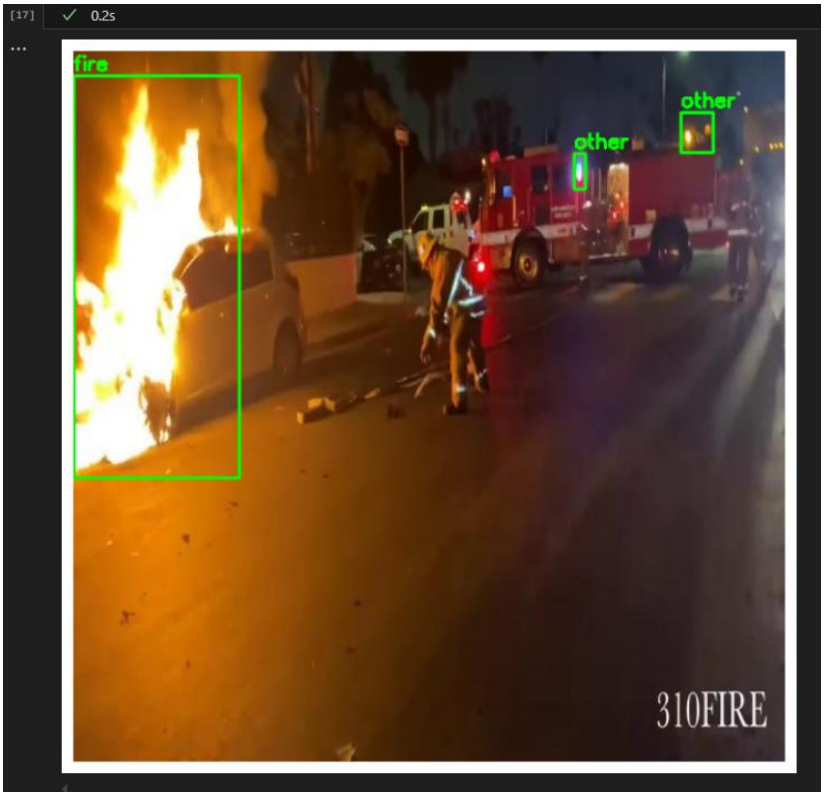

Рисунок 1.8 – Пример изображения с выделенными рамками bbox

1.5 Выявление потенциальных проблем

Как уже было отмечено в предыдущем разделе, набор данных содержит маленький процент «отрицательных» данных. То есть данных, не нацеленных на объекты возгорания и задымления.

Объем таких данных является критерием того, сможет ли модель корректно определять возгорания. Если в процессе обучения модель не взаимодействовала ни с какими объектами кроме огня и дыма, то, естественно ожидать, что она не сможет распознавать отсутствие искомых объектов.

По сути, «отрицательные» данные – это основной способ научить модель ничего не находить там, где ничего нет. Добавление достаточного числа отрицательных данных обеспечивает контроль ложных срабатываний, или же показатель FP (False Positive).

Выбранный датасет, несмотря на все свои преимущества, имеет недостаток отрицательных данных.

Тем не менее, данная проблема не является критической. Отрицательные данные не требуют разметки рамок, так как не содержат объекты, которые нужно распознавать. Поэтому для разнообразия нашего набора данных достаточно будет просто найти и внедрить в датасет определенное количество изображений реальных локаций, подверженных пожарной опасности.

Более того, структура COCO аннотаций позволяет легко внедрить такие изображения, просто добавив названия файлов в раздел images (добавлять данные в раздел annotations не придется, т.к. bbox для отрицательных данных не указывается).

Хоть датасет и предоставляет разметку «аномальных» данных, невозможно заранее гарантировать, что модели будет достаточно имеющихся признаков. Может понадобиться качественно новая информация. В таком случае хорошим решением будет использование аугментации.

Python библиотека Albumentation предоставляет широкую функциональность по аугментации данных. В том числе, и аугментацию «на лету» (on-the-fly), т.е. изменение и расширение обучающей выборки прямо во время обучения. Так что данный инструмент будем использовать во время обучения по мере необходимости.

1.6 Решение проблем

Как уже было отмечено, среди потенциальных проблем наиболее существенной является недостаток «отрицательных» данных. Поэтому в данном пункте попытаемся решить ее путем нахождения дополнительных данных.

Чтобы новые данные были релевантными, необходимо соблюдать требования предметной области. Наша модель нацелена на раннее распознавание и детекцию возгорания. Наиболее вероятным использованием модели будет встраивание в видеокамеру внутри или снаружи какого-либо помещения. Представимы также варианты использования на видеокамерах вышек и мониторинговых дронов.

Таким образом, предметная область определяет диапазон допустимых «отрицательных» значений. Это должны быть преимущественно изображения помещений, пространств рядом со зданиями, либо же съемок камер. При этом изображения не должны содержать возгораний.

На Kaggle были найдены два подходящих датасета: Smoke-Fire-Detection-YOLO – уже анализировавшийся датасет, не очень подходящий для задачи детекции возгорания, но подходящий для создания релевантного шума. И отличный датасет – Indoor Scenes от MIT, содержащий изображения помещений разного рода.

Была выделена выборка в 5000 изображений. Такое количество взято в целях балансировки предполагаемого качества распознавания будущей

модели. Объем полезных данных из нашего датасета составляет 21 тысячу изображений. А «отрицательных» примеров будет 5 / 26 ≈ 0,19 = 19%. Такой сбалансированный объем позволит значительно снизить процент ложных срабатываний, но при этом не помешает модели выполнять свою целевую задачу.

Скрипт по внедрению «отрицательной» выборки в датасет и аннотации представлен на рис. 1.9.

```python
import os; import json; import random
from PIL import Image

negative_images_dir = 'negative_images'
dataset_root = 'Fire-smoke-dataset/'
splits=("train", "valid", "test")
ratios=(0.7, 0.2, 0.1)

all_imgs = [f for f in os.listdir(negative_images_dir)]
random.shuffle(all_imgs)
counts = [round(r * len(all_imgs)) for r in ratios]

split_images = {
    split: all_imgs[sum(counts[:i]):sum(counts[:i+1])]
    for i, split in enumerate(splits)
}

for split, filenames in split_images.items():
    img_dir = dataset_root + '/' + split
    ann_path = img_dir + "/_annotations.coco.json"

    with open(ann_path, "r", encoding="utf-8") as f:
        coco = json.load(f)

    images = coco["images"]
    next_id = max((img["id"] for img in images), default=0) + 1

    for fname in filenames:
        src_path = negative_images_dir + '/' + fname
        dst_path = img_dir + '/' + fname

        with Image.open(src_path) as img:
            img = img.convert("RGB")
            img = img.resize((640, 640), Image.BILINEAR)
            img.save(dst_path)

        images.append({
            "id": next_id,
            "file_name": fname,
            "width": 640,
            "height": 640,
            "license": 1
        })
        next_id += 1

    with open(ann_path, "w", encoding="utf-8") as f:
        json.dump(coco, f, indent=2)
```

[1] ✓ 44.8s

Рисунок 1.9 – Скрипт для внедрения отрицательных изображений

Приведем статистику нашего обновленного датасета (см. таблицу 2).

Таблица 2 – Статистика объемов расширенных данных по выборкам

Характеристика	**train**	**test**	**validation**
Кол-во изображений	16398	2749	7092
Аннотаций Smoke	12348	2466	4869
Аннотаций Fire	14153	2161	5986
Аннотаций Other	3796	804	2135
Аннотаций всего	30297	5431	12990
Изображений без аннотаций	4027	558	1206

1.7 Постановка задачи анализа данных

Так как рассматриваемая предметная область помимо определения факта возгорания запрашивает указание места на изображении, задача анализа данных формулируется как *многоклассовая детекция объектов* (multiclass object detection). Это означает, что модель должна научиться локализовать объекты возгорания/задымления при помощи ограничивающего прямоугольника – bbox, а также классифицировать их.

На языке данных задачу можно описать следующим образом:

Вход модели – изображение фиксированного размера (скажем, 640x640).

Выход модели – список найденных объектов, каждый из которых представлен координатами bbox ([x, y, width, height]) и вероятностью принадлежности к одному из классов fire, smoke, other (учитывая специфику датасета, класс other будет означать детекцию объекта, похожего на вредоносный огонь, но таковым не являющийся).

Ожидаемым результатом разработки модели является корректно работающая модель, что означает соответствие рамки детекции

действительному расположению объекта, а также правильное предсказание класса объекта. Ожидается особое внимание к минимизации FP показателя.

1.8 Предполагаемое архитектурное решение

На основании анализа предметной области и проведенной предобработки датасета можно выделить несколько возможных архитектурных решений. Рассмотрим два наиболее подходящих.

Архитектура YOLO (You Only Look Once) известна своей высокой производительностью, компактностью и пригодностью для решения задач в реальном времени (соответствует нашему требованию детекции возгорания на ранней стадии).

Архитектура RetinaNet традиционно устойчива к классовому дисбалансу данных (благодаря использованию функции потерь Focal Loss). Также позволяет работать с разными базовыми сетями (backbone).

Тем не менее, исследование [5] показывает, что точность архитектуры YOLO в среднем выше, чем у RetinaNet.

Говоря о выборе инструмента для разработки модели, стоит сказать о предпочтении автора к использованию библиотеки Keras, в частности, Keras Functional API. В этом плане будет удобнее описывать архитектуру RetinaNet с подробной реализацией детекции.

Окончательное решение по архитектуре будет принято непосредственно перед началом разработки модели. Но на данный момент архитектура RetinaNet выглядит лучшим вариантом в контексте приобретения практического опыта по структурной разработке модели детекции.

ЗАКЛЮЧЕНИЕ

В ходе выполнения практики была рассмотрена предметная область детекции возгораний на изображениях с применением методов компьютерного зрения. Анализ показал существующие сложности, связанные с возможными аномалиями и неоднозначностью трактовки объекта огня.

Были рассмотрены существующие подходы к расширению и предварительной обработке данных. В частности, генерация синтетических данных и методы аугментации.

В результате поиска и учета релевантности среди множества открытых наборов данных был выбран наиболее подходящий под поставленную задачу детекции. Проведен детальный количественный анализ данных с элементами статистики, позволивший сделать заключение о недостатке в датасете «отрицательной» выборки. С целью повышения устойчивости модели к ложным срабатываниям вручную была добавлена выборка отрицательных релевантных изображений, доля которых по итогу составила 19% от общего объема.

В качестве возможного способа дальнейшей доработки набора данных предложена аугментация средствами Albumentation. Метод будет применяться в зависимости от качества обучения модели.

На основании результатов предобработки данных задача анализа была сформулирована как многоклассовая детекция объектов, включающая подзадачи локализации и классификации. Был проведен обзор возможных архитектурных решений для модели. Подходящими решениями стали YOLO и RetinaNet. Предпочтение было отдано второй из них.

Таким образом, можно сделать вывод о достижении поставленной цели по подготовке и анализу набора данных для обучения модели детекции возгорания. В процессе выполнения практики успешно решены задачи анализа

предметной области, обзора инструментария подготовки данных, поиска наборов данных, статистического описания данных с указанием проблем.

Полученные результаты создают неплохую основу для перехода к разработке и обучению модели в рамках выпускной квалификационной работы.

СПИСОК ИНФОРМАЦИОННЫХ ИСТОЧНИКОВ

1. Савченко А.А. Комплексный статистический анализ причин пожаров и их экономических последствий для развития региона: магистерская диссертация [Электронный ресурс] / Тольятти: ТГУ, 2024. – 91 с. – URL: https://dspace.tltsu.ru/bitstream/123456789/29917/1/Савченко%20А.А._ТБмд-2104а.pdf (дата обращения: 27.05.2025).

2. Kim H., Lee S., Park J. Early Fire Detection System by Using Automatic Synthetic Dataset Generation Model Based on Digital Twins [Электронный ресурс] // ResearchGate. – 2024. – URL: https://www.researchgate.net/publication/378409125_Early_Fire_Detection_System_by_Using_Automatic_Synthetic_Dataset_Generation_Model_Based_on_Digital_Twins (дата обращения: 28.05.2025).

3. Albumentations. Official Documentation: [Электронный ресурс]. URL: https://albumentations.ai/docs/ (дата обращения: 28.05.2025).

4. Lin T.-Y., Maire M., Belongie S. et al. Microsoft COCO: Common Objects in Context [Электронный ресурс] // arXiv. – 2014. URL: https://arxiv.org/pdf/1405.0312 (дата обращения: 29.05.2025).

5. Nife N. I., Chtourou M. A Comprehensive Study of Deep Learning and Performance Comparison of Deep Neural Network Models (YOLO, RetinaNet) [Электронный ресурс] // International Journal of Online and Biomedical Engineering (iJOE). – 2023. – Vol. 19, № 12. – P. 76. – URL: https://online-journals.org/index.php/i-joe/article/view/42607/13795 (дата обращения: 29.05.2025).