***Д. К. Зитцер, К. В. Радаев***
*Тюменский государственный университет, г. Тюмень*
**УДК 004.94**

# СИСТЕМА ПОДГОТОВКИ И РАЗМЕТКИ ДАННЫХ, ПОЛУЧЕННЫХ ИЗ СЕРВИСА IVIDEON TV

**Аннотация.** В статье представлено описание процесса разработки системы подготовки и разметки данных для задач компьютерного зрения. Описано сравнение существующих решений по сбору данных, и проведено их сравнение с созданной системой разметки данных.

**Ключевые слова**: Конвейер сбора данных, Apache Airflow, разметка данных, машинное обучение, компьютерное зрение.

**Введение.** Для успешного обучения алгоритмов машинного зрения необходимы данные. Получение и подготовка данных являются важной частью процесса разработки и исследования этих алгоритмов, определяя их точность. Некоторые из способов по автоматизации процесса сбора данных были рассмотрены в работах [1-3].

В небольших компаниях, процесс сбора выполняется разработчиками, работа которых высокооплачиваема [6], и тратить их время на такую рутинную задачу нецелесообразно. Усугубляет данную ситуацию и то, что все этапы сбора данных, выполняются разработчиками вручную:

1. Поиск видеоматериала, подходящий под условия съемки у заказчика;
2. Его скачивание для последующей обработки;
3. Подготовка данных к разметке:

a. Отбрасывание части материала, не содержащего в себе целевого объекта или действия;

b. Подготовка программной инфраструктуры для последующего аннотирования данных;

4. Разметка данных;
5. Перепроверка разметки данных.

В качестве примера приведу время затрачиваемое мной на подготовку набора, состоящего из 500 изображений и аннотаций к ним для задачи анализа пешеходного трафика (табл. 1).

*Таблица 1*

**Описание этапов ручного сбора с потраченным временем**

| *Название шага* | *Срок выполнения* | *Примечание* |
|---|---|---|
| Поиск источников | 2 часа | Поиск видеоматериала с подходящими заказчику условиями съемки |
| Выгрузка данных | 1 час | Время напрямую зависит от пропускной способности сети и количества материала |
| Подготовка данных к разметке | 1 час | Прямая зависимость от количества видеоматериала |
| Разметка изображений | до 8 часов | Прямая зависимость от количества изображений и объектов на них |
| Проверка данных | 1 час | Для исключения человеческого фактора |

Можно сделать вывод: разработчик потеряет до двух дней своего рабочего времени. В этой статье рассмотрим процесс создания конвейера по подготовке данных для обучения и тестирования нейронных сетей в контексте алгоритмов машинного зрения, эффективность которого была неоднократно доказана в работах [3-5].

**Проблема исследования.** Основной проблемой в процессе сбора данных, является высокая вовлеченность разработчика в рутинные шаги: найти, скачать, переместить, переименовать, создать проект, … В связи с описанным выше, возникает потребность в создании системы, которая позволит автоматизировать запуск и контроль конвейера.

**Материалы и методы.** Исходя из шагов, необходимых для сбора данных ручным способом, мы можем выявить основные этапы, которые могут быть полностью автоматизированы, к таким этапам мы можем отнести:

1. Выгрузка данных.
2. Фильтрация данных.
3. Предварительная разметка [7-8].

Как было (рис. 1).


*Рис. 1.* Существующий способ получения набора данных

Как будет (рис. 2).

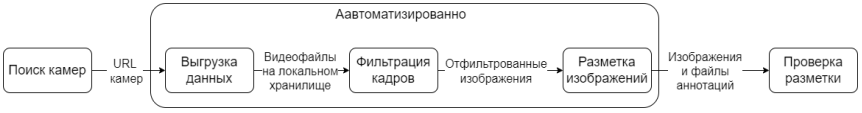


*Рис. 2.* Будущий способ получения набора данных

В силу доступности и разнообразия данных в качестве источника видеоматериала был взят сервис Ivideon TV [9]. К тому же, сервис имеет общедоступный API, благодаря которому материал будет выгружаться автоматически.

Входные данные:

1. Список URL-адресов камер с которых, нужно выгрузить видео.
2. Путь для сохранения скачанного видео.

Выходные данные:

1. Видеофайлы с расширением .flv.

Алгоритм преобразования входных данных в выходные:

1. Формируется GET-запрос к открытому API сервиса Ivideon TV.
2. Запускаются параллельные процессы по скачиванию.
3. Логирование статуса начала загрузки.
4. Ожидание завершения загрузки.
5. Логирование статуса завершение загрузки и продолжительности скачивания.
6. Завершение работы модуля.

Для разработки видеоаналитики или обучения нейронной сети, нас интересуют только те кадры, на которых присутствуют пешеходы, люди или другие необходимые объекты. Но, как показывает практика, при покадровом анализе большую временную часть скачанного видео может занимать сцена без какого-либо движения, то есть статичная. Модуль фильтрации как раз предназначен для отбрасывания статичных кадров.

Входные данные:

1. Область интереса в кадре;

2. Пороговые значения для сравнения соседних кадров между собой;

3. Видеофайл, кадры из которого требуется отобрать.

Выходные данные:

1. Кадры, в которых присутствуют изменения в интересующей области в сравнении с предыдущими.

Алгоритм преобразования входных данных в выходные:

1. Захват переданного видео
2. Последовательное считывание кадра
3. Выделение из кадра области интереса
4. Сравнение кадров между собой

a. если кадры сильно отличаются, сохранить

b. если кадры не отличаются, пропустить

5. Завершение работы модуля.

Идея данного этапа заключается в том, чтобы провести предварительную разметку данных при помощи open source моделей. Они не покажут высокое качество разметки на наших данных, однако даже малая часть корректных аннотаций позволит сократить трудозатраты будущих асессоров. Существует достаточное количество “зоопарков” open source моделей, выбор был остановлен на моделях детекции персон из коллекции OpenVINO.

Модуль пред разметки будет иметь следующую структуру:

Входные данные:

1. Модель машинного зрения для аннотирования изображений
2. Изображения
3. Пороговое значение уверенности модели

Выходные данные:

1. Файлы аннотация сохраненные в заданную директорию

Алгоритм преобразования входных данных в выходные:

1. Считывание изображения
2. Передача входных данных в нужном формате нейронной сети
3. Постпроцессинг результатов

4. Приведение и сохранение полученных координат ограничивающих рамок, к необходимому формату (в нашем случае таким формат YOLO)

5. Завершение работы модуля.

В нашем случае, для проверки разметки данных был выбран, сервис Yandex.Toloka, к преимуществам данного сервиса можно отнести:

1. Большое количество людей готовых проверить наши данные за небольшую плату.

2. Возможность создать гибко конфигурируемый интерфейс для проверки аннотаций.

3. Активное сообщество, готовое помочь с любыми возникающими вопросами

4. Открытый API, позволяющий загружать данные, в автоматическом режиме.

Таким образом, нам нужно лишь обратиться к API, чтобы передать данные на загрузку и ожидать когда задания будут выполнены.

Примечание: Для того, чтобы загрузить данные на проверку изображения должны находится в сети интернет, и должны существовать ссылки до этих изображений. В случае наших данных для хранения использовалось Yandex Object Storage, но это может быть любой другой сервис предоставляющий такой же функционал.

По отдельности каждый из описанных этапов не может работать корректно, нужно настроить передачу информации и данных между ними. В общем случае, взаимодействие между модулями можно описать с помощью диаграммы (рис. 3).

Но для реализации такого взаимодействия между модулями, нужно использовать соответствующий инструмент.

Ниже приведена таблица сравнения нескольких популярных сервисов для управления рабочими процессами и планирования задач (табл. 2).

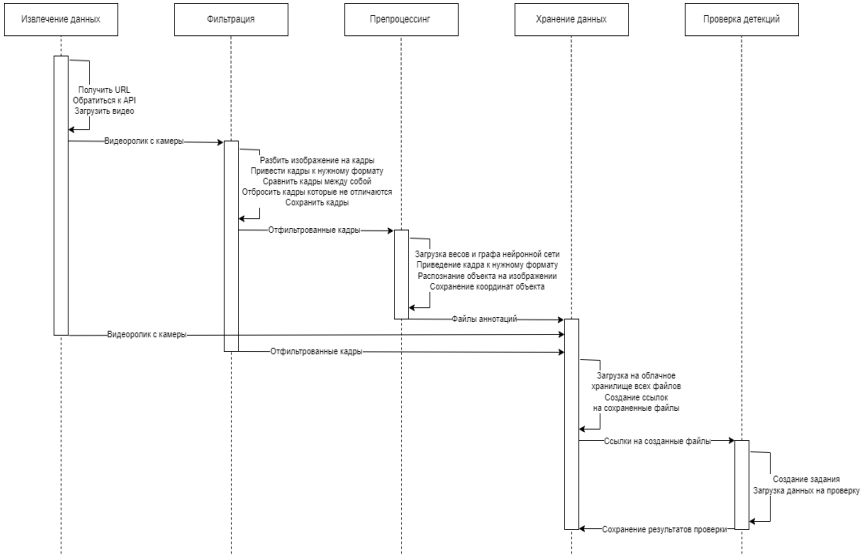


*Рис. 3.* Описание взаимодействия модулей

*Таблица 2*

**Сравнение сервисов управления рабочими процессами**

| *Название* | *Создан* | *Способ определения рабочих процессов* | *Написан при помощи* | *Планирование* | *Пользовательский интерфейс* | *Платформа* |
|---|---|---|---|---|---|---|
| Airflow | Airbnb | Python | Python | + | + | Любая |
| Argo | Applatix | YAML | GO | - | + | Kubernetes |
| Azkaban | Linkedin | YAML | JAVA | + | + | Любая |
| Conductor | Netflix | JSON | JAVA | - | + | Любая |
| Luigi | Spotify | Python | Python | - | + | Любая |
| Metaflow | Netflix | Python | Python | - | + | Любая |
| Nifi | NSA | Пользовательский интерфейс | JAVA | + | + | Любая |
| Oozie | - | XML | JAVA | + | + | Любая |

Основными критериями, на которые обращалось внимание при выборе инструмента являются:

1. Возможность планирования запусков
2. Управление запусками, отслеживание статуса и результата в графическом интерфейсе
3. Определение рабочих процессов при помощи Python

Исходя из таблицы приведенной ниже видно что только один инструмент удовлетворяет всем перечисленным критериям — это Apache Airflow [10].

Все процессы в AirFlow, определяются с помощью языка программирования Python и есть возможность работать с Docker контейнерами, что позволит избежать проблем с установкой зависимостей у разных пользователей и настроить запуск каждого модуля в своем независимом окружении с предустановленными пакетами (рис. 4).

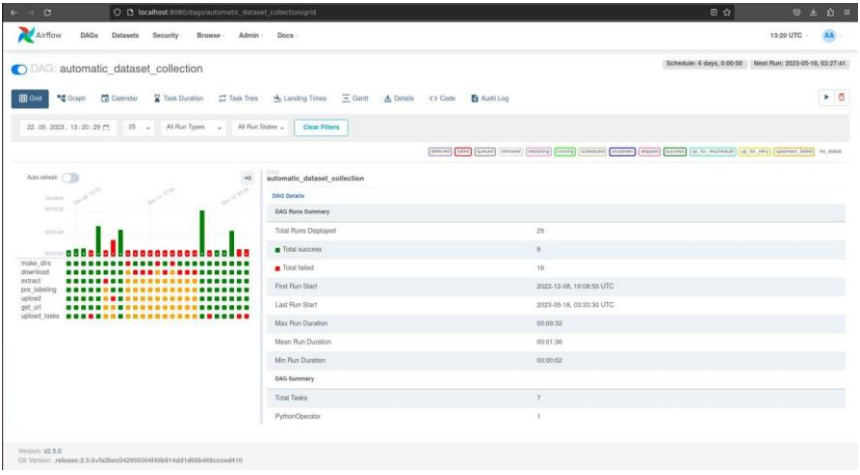


*Рис. 4.* Пользовательский интерфейс Apache Airflow

Работа с Apache AirFlow заключается в написании DAG (Ориентированный ациклический граф) файла. Простыми словами DAG это сущность, объединяющая ваши задачи в единую цепочку задач, где явно видны зависимости между узлами.

Порядок работы с конвейером:

1. Переходим на сайт сервиса Ivideon TV
2. Ищем необходимые для нас камеры
3. Копируем URL ссылку до найденной камеры/камер
4. Открываем пользовательский интерфейс AirFlow
5. Настраиваем DAG, и передаем ссылки до найденных камер
6. Запускаем DAG
7. Ожидаем окончания выполнения конвейера
8. Переходим на сайт сервиса Yandex Toloka
9. Ожидание проверки и корректировки аннотаций внешними асессорами
10. Выгружаем/Используем полученный набор данных

В конце всех выполненных работ, у нас имеется два способа сбора данных: Ручной и Полуавтоматический. В таблице (табл. 3) можно увидеть сравнение данных способов.

*Таблица 3*

**Сравнение методов сбора данных**

| *Название этапа* | *Полуавтоматический* | *Ручной* |
|---|---|---|
| *1* | *2* | *3* |
| Поиск источников | 1-2 часа | 1-2 часа |
| Примечание к этапу “Поиск источников” | - | - |
| Выгрузка данных | 0 минут | 20-40 минут |
| Примечание к этапу “Выгрузка данных” | Автоматизирован | - |
| Фильтрация кадров | 0 минут | 1 час |
| Примечание к этапу “Фильтрация кадров” | Автоматизирован | Нет, одного определенного шаблона по которому будут фильтроваться кадры, из-за чего процесс отнимает много времени у разработчика |
| Разметка изображений | 0 минут | 5-8 часов |
| Примечание к этапу “Разметка изображений” | Автоматизирован | - |

*Окончание табл. 3*

| *1* | *2* | *3* |
|---|---|---|
| Проверка данных | 3-4 часа | 1 час |
| Примечание к этапу “Проверка данных” | Из-за того что человек делает более точные аннотации чем нейронная сеть, приходится затрачивать больше времени на исправлении детекций | - |

Исходя из данной таблицы был сформулирован вывод, что единственным этапом в новом способе сбора данных на котором, затрачивается больше времени, чем на старом, является проверка данных, но данный минус полностью компенсируется, экономией времени на предыдущих этапах. В общем, сравнение этих двух способов показывает выигрыш, нового способа в ~5 часов, для сбора набора данных в 500 изображений, но стоит не забывать тот факт, что чем больше будет требоваться набор данных тем больше будет расти разница во времени между наборами.

Была разработана система позволяющая упростить, процесс подготовки данных, снизить время для сбора одного датасета. Так же к преимуществам системы можно отнести то, что она является расширяемой, и при возникновении необходимости ее расширить это возможно сделать с легкостью.

## СПИСОК ЛИТЕРАТУРЫ


1. Бесшапошников Н. О. Автоматизация разметки набора данных для нейронных сетей / Н. О. Бесшапошников, М. А. Кузьменко, А. Г. Леонов, М. А. Матюшин // Cyberleninka: [сайт]. — URL: https://cyberleninka.ru/article/n/ avtomatizatsiya-razmetki-nabora-dannyh-dlya-neyronnyh-setey/viewer (дата обращения: 29.05.2023). — Текст: электронный.
2. Меньшиков Я. С. Преимущества автоматического сбора данных в сети интернет над ручным сбором данных / Я. С. Меньшиков // Cyberleninka: [сайт]. — URL: https://cyberleninka.ru/article/n/preimuschestva-

avtomaticheskogo-sbora-dannyh-v-seti-internet-nad-ruchnym-sborom-dannyh/viewer (дата обращения: 29.05.2023). — Текст: электронный.
3. Халяфиев Р. А. Сбор данных для разработки нейронной сети / Р. А. Халяфиев // Cyberleninka: [сайт]. — URL: https://cyberleninka.ru/article/n/sbor-dannyh-dlya-razrabotki-neyronnoy-seti/viewer (дата обращения: 29.05.2023). — Текст: электронный.
4. Флегонтов А. В. Система интеллектуальной обработки данных / А. В. Флегонтов, В. В. Фомин // Cyberleninka: [сайт]. — URL: https://cyberleninka.ru/article/n/sistema-intellektualnoy-obrabotki-dannyh/viewer (дата обращения: 29.05.2023). — Текст: электронный.
5. Гилязев Р. А. Активное обучение и краудсорсинг: обзор методов оптимизации разметки данных / Р. А. Гилязев, Д. Ю. Турдаков // Cyberleninka: [сайт]. — URL: https://cyberleninka.ru/article/n/aktivnoe-obuchenie-i-kraudsorsing-obzor-metodov-optimizatsii-razmetki-dannyh/viewer (дата обращения: 29.05.2023). — Текст: электронный.
6. Зарплаты в ИТ. По всем ИТ-специализациям // ХАБР [сайт]. — URL: https://career.habr.com/salaries (дата обращения 29.05.2023) — Текст: электронный.
7. Разметка данных в машинном обучении: процесс, разновидности и рекомендации // [сайт]. — URL:https://habr.com/ru/articles/678524/ (дата обращения 29.05.2023) — Текст: электронный.
8. Что такое аннотация данных 2023? (Лучшие инструменты, типы, проблемы, тенденции): [сайт]. — URL:https://ru.shaip.com/blog/the-a-to-z-of-data-annotation/ (дата обращения 29.05.2023) — Текст : электронный.
9. Облачный сервис для видеонаблюдения Ivideon. // [сайт]. — URL:https://ru.ivideon.com/ (дата обращения 29.05.2023). — Текст : электронный.
10. Официальная документация Apache Airflow. // [сайт]. — URL:https://airflow.apache.org/docs/ (дата обращения 29.05.2023) — Текст : электронный.