МИНОБРНАУКИ РОССИИ
Федеральное государственное бюджетное образовательное учреждение
высшего образования
«САРАТОВСКИЙ НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ
ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ ИМЕНИ Н.Г.ЧЕРНЫШЕВСКОГО»

Кафедра информатики и программирования

РАЗРАБОТКА МАСШТАБИРУЕМОГО WEB-ПРИЛОЖЕНИЯ ДЛЯ РАСПОЗНАВАНИЯ СОДЕРЖИМОГО ТАБЛИЦ В PDF ФОРМАТЕ

АВТОРЕФЕРАТ БАКАЛАВРСКОЙ РАБОТЫ

студента 4 курса 441 группы
направления 02.03.03 Математическое обеспечение и администрирование информационных систем
факультета компьютерных наук и информационных технологий
Щурова Дмитрия Ивановича

Научный руководитель:
д.ф.-м.н., профессор _____ Андрейченко Д.К.

Зав. кафедрой МОВКИС,
д.ф.-м.н., профессор _____ Андрейченко Д.К.

Саратов 2022

ВВЕДЕНИЕ

Актуальность темы. Для современного мира использование информационных технологий стало повседневным занятием. Любой человек, у которого есть доступ в интернет может найти практически любую информацию, которая ему может понадобиться для повседневной жизни или работы. За время существования информационных технологий было создано большое количество вспомогательных систем, которые упрощают и улучшают жизнь каждого человека. Все это представляет собой некоторую автоматизацию простых рутинных процессов или сложных задач, которые требуют больших вычислительных мощностей.

Несмотря на многообразие различных информационных систем, некоторые рутинные задачи все равно приходится выполнять человеку. Некоторые из них не автоматизированы из-за сложности переноса человеческого мышления в информационные системы, а некоторые и вовсе нет необходимости автоматизировать.

Задача распознавания текста и таблиц в частности является по прежнему актуальной на текущий момент. Многие люди вручную анализируют и переносят данные с таблиц на фото или файлов перепечатывая содержимое вручную, не имея возможности его скопировать из-за отсутствия текстового слоя в большинстве файлов. Такого рода деятельность можно считать рутинной. Рутинную деятельность чаще всего можно автоматизировать или хотя бы ускорить.

Уже существует некоторое количество бесплатных сервисов, предоставляющих подобные услуги. Каждый из них работает на удаленном сервере и не позволяет получить доступ к их исходному коду. Данный факт автоматически закрывает возможность их использования на собственных вычислительных мощностях и как следствие использования в собственных проектах.

В данной работе была поставлена цель реализовать систему, которая позволила бы ускорить подобного рода деятельность и предусмотреть

возможность горизонтального масштабирования данного решения. Также данная система будет иметь открытый исходный код, что позволит использовать его в собственных проектах любого масштаба.

Цель бакалаврской работы – Разработка сервиса, позволяющего производить преобразование таблиц с фотографий в структуру данных, которая позволит производить обработку информации программным путем.

Поставленная цель определила **следующие задачи**:

- Рассмотреть существующие системы с похожим функционалом
- Определить необходимый набор средств и технологий для разработки собственного сервиса, которые позволят достигнуть цель работы
- Изучить выбранным набор средств и технологий, необходимых для достижения цели работы
- Используя полученные знания реализовать собственный сервис, позволяющий производить обработку подобных файлов
- Предусмотреть возможность горизонтального масштабирования сервиса, для распознавания таблиц

Методологические основы обработки изображений и разработке web-сервисов представлены в работах Шакла Нишант, Конушин.А.С, А.М.Миронов, Мартин Клеппман, Е. Н. Десятирикова, Хадж Али Муса, Ходар Алмосана, Алькади Усама, Раджаб Хаян.

Практическая значимость бакалаврской работы. Реализованная система позволяет производить обработку документов, состоящих из таблиц. Данное решение может быть развернуто на любой из систем, обладает хорошими показателями масштабируемости. Данная система может использоваться, например, для упрощения обработки сканов финансовых выписок и отчетов при выдаче кредитов юридическим лицам.

Структура и объём работы. Бакалаврская работа состоит из введения, 3 разделов, заключения, списка использованных источников и 2 приложений. Общий объем работы – 62 страниц, из них 43 страниц – основное содержание,

включая 10 рисунков и 1 таблицы, список использованных источников информации – 20 наименований.

КРАТКОЕ СОДЕРЖАНИЕ РАБОТЫ

Первый раздел «Основные понятия» посвящен определению основных понятий необходимых для выполнения работы. В данной главе рассматриваются основные понятия, связанные с машинным обучением и концепции, связанные с ним. Должное внимание также уделено концепциям и принципам компьютерного зрения.

Рассмотрены основные принципы контейнеризации приложений и возможности произвести ее при помощи Docker.

Были также описаны принципы масштабирования приложений и инструменты, которые будут использоваться в процессе реализации приложения.

Второй раздел «Описание реализованной системы» посвящен подробному разбору реализованной системы, описанию полученной архитектуры и инфраструктуры.

Архитектура приложения представляет собой набор сервисов, которые в сумме позволяют извлекать информация из таблиц, находящихся на изображение.

Микросервисная архитектура, используемая в приложение, позволяет сделать его более гибким и устойчивым к сбоям. Предоставляет возможность точечно увеличивать производительность каждого компонента. Очень важным фактом при выборе архитектуры была необходимость иметь возможность легко и быстро разрабатывать новый функционал, что хорошо вписывается в микросервисную архитектуру.

Ниже представлена схема, на которой представлены все компоненты приложения (Рисунок 1).

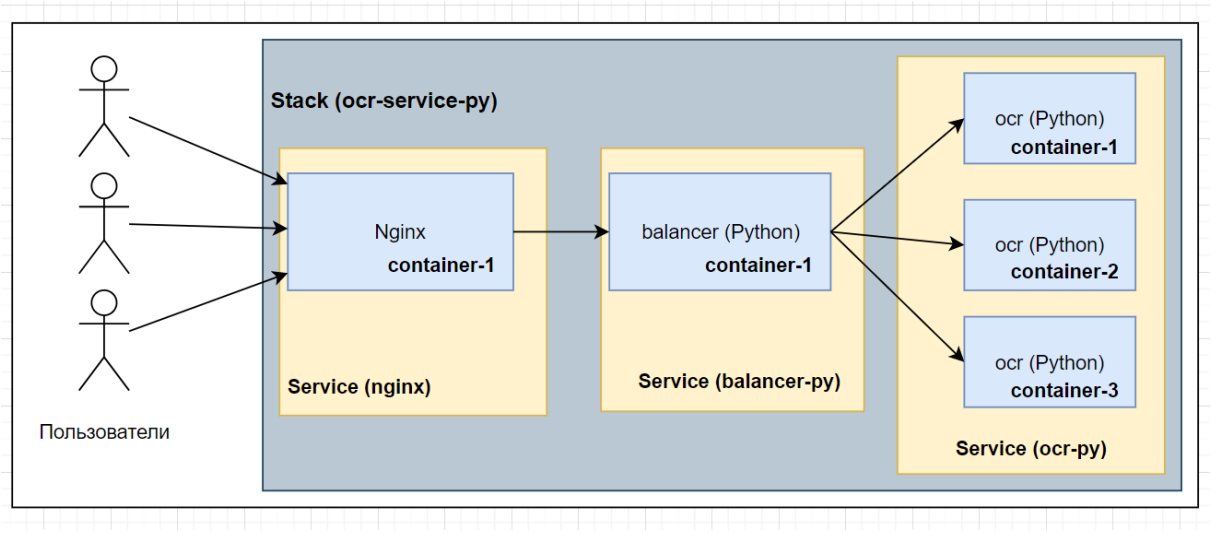

Рисунок 1 – Структура приложения

Основным компонентом приложения является сервис ocr-py. В нем происходит само извлечение текста из изображения. Оно и является основным потребителем вычислительной мощности. Также этот сервис является, единственным, который подразумевает горизонтальное масштабирование в рамках данной работы.

Сервис balancer-py представляет собой инструмент для разбиения большого файла на более мелкие по средствам разделения многостраничного документа на документы, состоящие из одной страницы. Это позволяет равномерно распределить нагрузку внутри сервиса ocr-py.

Сервис nginx в данном случае выступает в качестве обратного прокси, чтобы быть единственной точкой входа в систему. Такой подход позволяет обеспечить лучшую безопасность и надежность системе. Кроме всего этого это добавляет всей системе еще точку роста для дальнейшего развития и увеличения вычислительных мощностей.

Третий раздел «Демонстрация работоспособности системы» посвящен проведенным замерам и примерам использования реализованной системы. Основной акцент сделать на разбор примера результата и показателям при увеличении числа одновременных запросов и числа одновременно работающих экземпляров сервиса на компьютере.

При распознавании таблицы, которая представлена на Рисунке 2, был получен следующий результат.

Информация из Государственного информационного ресурса бухгалтерской (финансовой) отчетности (Ресурса БФО)	
Дата формирования информации	23.05.2022
Номер выгрузки информации	№ 0710099_6451105825_2021_000_20220523_52ac553a-503e-49b3-bd6f-0c09e67b6a2f
Настоящая выгрузка содержит информацию о юридическом лице:	
Полное наименование юридического лица	ООО ЗАВОД САРАТОВГАЗАВТОМАТИКА
включенная в Государственный информационный ресурс бухгалтерской (финансовой) отчетности по состоянию на 23.05.2022	
ИНН	6451105825
КПП	645101001
Код по ОКПО	00153672
Форма собственности (по ОКФС)	16
Организационно-правовая форма (по ОКОПФ)	
Вид экономической деятельности по ОКВЭД 2	26.51.5
Местонахождение (адрес)	410008, Саратовская обл, г Саратов, ул Лопатина Гора, 7
Единица измерения	Тыс. руб.
Бухгалтерская отчетность подлежит обязательному аудиту	Да
Наименование аудиторской организации/ФИО индивидуального аудитора	
ИНН	
ОГРН/ОГРНИП	

Рисунок 2 – Пример таблицы

[[

"Информация из Государственного информационного ресурса бухгалтерской (финансовой) отчетности (Ресурса БФО)"]

,[""],[

"Дата формирования информации", "23.05.2022"],[

"Номер выгрузки информации", "№ 0710099 _6451105825_2021_000_20220523_52ac553a- 503e-4953-Б46Е-0c09eб7ьба?Е"],[

"Настоящая выгрузка содержит информацию о юридическом лице:"],[

"Полное наименование юридического лица", "ООО ЗАВОД САРАТОВГАЗАВТОМАТИКА"], [

""],[""],[

"ИНН", "6451105825"],[

"", "645101001"],[

"Код по ОКПО", "00153672"],[

"Форма собственности (по ОКФС)", "16"],[

"Организационно-правовая форма (по ОКОПФ)", ""],[

"Вид экономической деятельности по ОКВЭД 2", "26.51.5"],[

"Местонахождение (адрес)", "410008, Саратовская обл, г Саратов, ул Лопатина Гора, 7"],[

"Единица измерения", "Тыс. руб."],[

"Бухгалтерская отчетность подлежит обязательному аудиту", ""],[

"Наименование аудиторской организации/ФИО индивидуального аудитора", ""],[

"ИНН", ""],[

"ОГРН/ОГРНИП", ""

]]

Также в данной главе уделено тому как запускается приложение. Для этого используются либо команды Docker в командной строке или терминале,

либо графическая оболочка для этих целей. В работе в качестве примера графической оболочки выступает PyCharm.

В таблице 1 представлены результаты масштабирования на системе со следующей конфигурацией:

1. CPU: «Intel® Core™ i5-10400F × 12» - Количество ядер 6. · Количество потоков 12. Режим Turbo не использовался.
2. GPU: «NVIDIA GeForce RTX™ 3060»
3. RAM: «32Gb»
4. ОС: «Manjaro Linux x64»

По результатам замеров видно, что если количество документов близко к количеству ядер в системе, то достигается максимально эффективное использование ресурсов процессора системы. В процессе тестирования загрузка процессора при количестве обрабатываемых страниц равным 6 была достигнута практически пиковая нагрузка на процессор (примерно 90% на каждом из ядер).

При дальнейшем увеличение количества одновременно обрабатываемых страниц в системе время обработки всего документа стремительно увеличивается, что говорит о снижение эффективности.

Таблица 1 - Результаты масштабирования

Количество запросов разрешенных на каждый из контейнеров	Количество контейнеров в балансировке	Результат обработки end-to-end (мин)
1	1	8.1
1	2	4.2
1	4	2.4
1	6	1.7
1	8	1.75
1	10	4.9
2	2	2.4
2	3	1.7
2	4	1.8
3	1	3
3	2	1.8
3	3	2.1
5	1	2
6	1	1.8
8	1	1.95
10	1	9.1

ЗАКЛЮЧЕНИЕ

В результате проделанной работы был разработано легко масштабируемое приложение, позволяющее производить обработку графических файлов в формате pdf в удобную для обработки структуру данных. В качестве основных инструментов были использованы Python, Pdf2Image, OpenCV и Tesseract-OCR для написания логики обработки файлов. Для организации масштабируемости системы были использованы Docker, Docker Compose и Docker Swarm. В качестве балансировщика нагрузки и

обратного-proxy был использован Nginx и встроенные инструменты Docker Swarm.

Основные источники информации:

1. Машинное обучение часть 1 А.М.Миронов Московский Государственный Университет Механико-математический факультет Кафедра математической теории интеллектуальных систем.
2. Компьютерное зрение - Конушин.А.С - ВМК МГУ - 2019 г.
3. Шакла Нишант - Машинное обучение и TensorFlow - СПБ.: 2019. - 336с.
4. Мартин Клеппман Высоконагруженные приложения. Программирование, масштабирование, поддержка - 2017г.
5. Балансировка нагрузки в облачных вычислениях Е. Н. Десятирикова, Хадж Али Муса, Ходар Алмосана, Алькади Усама, Раджаб Хаян 2017г.
6. Сэмми Пьюривал - Основы разработки веб-приложений.