

ПРИМЕНЕНИЕ МИКРОСЕРВИСНОЙ АРХИТЕКТУРЫ ПРИ АНАЛИЗЕ МЕДИЦИНСКИХ ИЗОБРАЖЕНИЙ

Богданов А.А.,

студент 4 курса ВГУ имени П.М. Машерова, г. Витебск, Республика Беларусь

Научный руководитель – Ермоченко С.А., канд. физ.-мат. наук, доцент

Современная медицина активно использует цифровые технологии для диагностики и лечения, что приводит к стремительному росту объема данных, особенно медицинских изображений, таких как рентгенограммы, МРТ и КТ. Обработка этих данных требует высокой вычислительной мощности, гибкости и скорости, что традиционные монолитные системы часто не могут обеспечить из-за их ограниченной масштабируемости и сложности обновления [1]. В этом контексте микросервисная архитектура становится перспективным подходом, позволяющим разделить сложные процессы анализа изображений на независимые, легко управляемые компоненты, каждый из которых выполняет конкретную задачу – от предварительной обработки до интерпретации с использованием искусственного интеллекта (ИИ).

Микросервисы предлагают модульность, упрощают интеграцию новых алгоритмов и технологий, а также обеспечивают параллельную обработку данных, что особенно важно для анализа медицинских изображений в реальном времени. Например, в условиях высокой нагрузки, таких как массовый скрининг или экстренная диагностика, микросервисная архитектура позволяет быстро масштабировать отдельные компоненты системы, минимизируя затраты и повышая эффективность [2]. Цель данного доклада – рассмотреть, как применение микросервисной архитектуры оптимизирует анализ медицинских изображений и какие преимущества это дает в клинической практике.

Материал и методы. Для анализа были изучены существующие подходы к обработке медицинских изображений с использованием микросервисной архитектуры, включая исследования из PubMed и IEEE Xplore, а также примеры реализации в современных платформах, таких как AWS и Kubernetes [3]. Рассматривались микросервисы для типичных этапов обработки изображений: загрузка и хранение данных (с использованием облачных решений), предварительная обработка (шумоподавление, нормализация), сегментация (выделение областей интереса с помощью ИИ), классификация (диагностика с применением сверточных нейронных сетей) и визуализация результатов.[4] Методы включали сравнение производительности микросервисных систем с монолитными на основе таких параметров, как время обработки, масштабируемость и отказоустойчивость.

Результаты и их обсуждение. Результаты показали, что микросервисная архитектура значительно ускоряет процесс анализа медицинских изображений за счет параллельного выполнения задач. Например, в модельной системе сегментация изображения на основе ИИ и его классификация выполнялись независимо, что сократило общее время обработки на 35% по сравнению с монолитным подходом [2]. Для демонстрации преимуществ был разработан прототип микросервиса, выполняющего сегментацию изображений (например, выделение опухолей на МРТ). Ниже приведен пример кода на Python с использованием Flask и OpenCV:

```
from flask import Flask, request, jsonify
import cv2
import numpy as np
from PIL import Image
import io

app = Flask(__name__)
```

```

@app.route('/segment', methods=['POST'])
def segment_image():
    try:
        file = request.files['image']
        img = Image.open(file.stream)
        img_array = np.array(img)
        gray = cv2.cvtColor(img_array, cv2.COLOR_RGB2GRAY)
        _, thresh = cv2.threshold(gray, 127, 255,
                                   cv2.THRESH_BINARY)
        result = Image.fromarray(thresh)
        output = io.BytesIO()
        result.save(output, format='PNG')
        output.seek(0)
        return jsonify({'status': 'success',
                        'message': 'Image segmented'}), 200
    except Exception as e:
        return jsonify({'status': 'error',
                        'message': str(e)}), 500

if __name__ == '__main__':
    app.run(host='0.0.0.0', port=5000, debug=True)

```

Этот микросервис принимает изображение через HTTP-запрос, выполняет бинаризацию для выделения областей и возвращает результат в формате JSON. Тестирование на 100 изображениях (512x512 пикселей) показало, что обработка занимает 12 секунд на одном ядре, а при масштабировании до 4 экземпляров – 4 секунды, что подтверждает эффективность параллельной обработки [3].

Масштабируемость оказалась ключевым преимуществом: при увеличении числа запросов (1000 изображений в час) микросервисы позволили динамически выделять ресурсы только для нагруженных компонентов, снизив затраты на 20–25%. Отказоустойчивость повысилась благодаря изоляции сервисов: сбой в одном модуле не влиял на другие. Однако сложность управления сервисами и их взаимодействие через API требовали оркестраторов, таких как Kubernetes, и облачных хранилищ (например, Amazon S3) [2]. Микросервисная архитектура особенно полезна для задач, требующих скорости и адаптивности, но для небольших клиник может быть избыточной из-за высоких затрат на внедрение [3].

Заключение. Микросервисная архитектура демонстрирует значительные преимущества в анализе медицинских изображений, включая ускорение обработки, гибкость масштабирования и повышение надежности системы. Она позволяет эффективно интегрировать технологии ИИ, адаптироваться к растущим объемам данных и обеспечивать бесперебойную работу в условиях высокой нагрузки [3]. Несмотря на сложности внедрения, такие как координация сервисов и первоначальные затраты, эти проблемы решаются с помощью современных инструментов. В перспективе микросервисы могут стать стандартом для медицинских систем, повышая качество и доступность диагностики на основе изображений.

1. Корчевская, Е.А. Интеллектуальный анализ ультразвуковых исследований / Е.А. Корчевская // Наука – образованию, производству, экономике [Электронный ресурс]: материалы 76-й Региональной научно-практической конференции преподавателей, научных сотрудников и аспирантов, Витебск, 1 марта 2024 г. – Витебск: ВГУ имени П. М. Машерова, 2024. – С. 37–38. – Библиогр.: с. 38 (1 назв.). Электронная версия – <https://rep.vsu.by/handle/123456789/42132>.

2. Ньюман, С. Микросервисы: проектирование высокомасштабируемых систем / С. Ньюман. – Санкт-Петербург: Питер, 2020. – 368 с.

3. Ричардсон, К. Микросервисные паттерны: проектирование масштабируемых систем / К. Ричардсон. – Москва: Вильямс, 2019. – 520 с

4. Корчевская, Е.А. Системы искусственного интеллекта: метод. рекомендации / Е.А. Корчевская, С.А. Ермоченко, М.Г. Семенов; М-во образования Республики Беларусь, Учреждение образования «Витебский государственный университет имени П. М. Машерова», Каф. прикладного и системного программирования. – Витебск: ВГУ имени П.М. Машерова, 2020. – 47, [1] с.: ил. – Библиогр.: с. 47. Электронная версия – <https://rep.vsu.by/handle/123456789/21350>.

ПРИМЕНЕНИЕ НЕЙРОСЕТИ ДЛЯ ПЕРВИЧНОГО АНАЛИЗА СОСТОЯНИЯ ПЕЧЕНИ ПО УЗИ

Бритик В.С.,

студент 4 курса ВГУ имени П.М. Машерова, г. Витебск, Республика Беларусь

Научный руководитель – Витько Е.А., канд. физ.-мат. наук

В настоящее время ведутся активные исследования по применению нейронных сетей для решения различных практических задач, в том числе, в сфере здравоохранения [1–3].

Целью работы является создание программного обеспечения для первичного анализа состояния печени пациента по УЗИ.

Материал и методы. Для создания программного обеспечения использовались следующие технологии: язык программирования Python, библиотека машинного обучения TensorFlow, инструмент PyDicom для обработки файлов медицинского формата, а также OpenCV для обработки изображений стандартных форматов.

Результаты и их обсуждение. Перед началом разработки были проанализированы несколько медицинских сайтов на наличие файлов УЗИ печени с разного рода заболеваний, а также УЗИ здоровой печени.

Разработанная программа для автоматизированного анализа состояния печени на основе УЗИ состоит из четырех основных частей: **data_loader**, **model_builder**, **train**, и **predict**. Модуль **data_loader** отвечает за загрузку и предварительную обработку данных. Он подготавливает изображения из различных форматов, таких как DICOM и JPG, масштабируя их и преобразуя в формат, пригодный для работы с нейросетью. Этот этап включает нормализацию данных и структурирование их в виде, подходящем для обучения модели.

Часть **model_builder** содержит описание архитектуры нейросети. Здесь создаются следующие слои: сверточные (Conv2D), пуллинговые (MaxPooling2D), полносвязные (Dense), которые формируют основу модели. Раздел **train** отвечает за обучение модели на подготовленных данных, используя заданные параметры, такие как функции активации и потерь. Наконец, модуль **predict** позволяет использовать обученную модель для анализа новых данных, возвращая вывод о состоянии печени (здоровая или больная). Эта структурированная организация программы обеспечивает её гибкость, масштабируемость и удобство в использовании.

Заключение. В результате проведенного исследования было разработано приложение на языке программирования Python, которое позволяет определить первичную стадию заболевания печени по состоянию её ткани, с возможностью обработки файлов формата DICOM, а также изображений в стандартных форматах.

1. Корчевская, Е.А. Интеллектуальный анализ ультразвуковых исследований / Е.А. Корчевская // Наука – образованию, производству, экономике [Электронный ресурс]: материалы 76-й Региональной научно-практической конференции преподавателей, научных сотрудников и аспирантов, Витебск, 1 марта 2024 г. – Витебск: ВГУ имени П. М. Машерова, 2024. – С. 37–38. URL: <https://rep.vsu.by/handle/123456789/42132> (дата обращения 01.03.2025).

2. Никонов, Н.Д. Использование сверточных нейронных сетей для решения задач классификации в неконтролируемых условиях / Н.Д. Никонов, Т.В. Никонова, О.Е. Рубаник, Е.А. Корчевская // Веснік Віцебскага дзяржаўнага ўніверсітэта імя П.М. Машэрава імя П.М. Машэрава. – 2023. – № 2. – С. 5–11. URL: <https://rep.vsu.by/handle/123456789/39417> (дата обращения 01.03.2025).