ИСКУССТВЕННЫЙ ИНТЕЛЛЕКТ И МАШИННОЕ ОБУЧЕНИЕ / ARTIFICIAL INTELLIGENCE AND MACHINE LEARNING

DOI: https://doi.org/10.60797/IRJ.2024.145.120

РАЗРАБОТКА ПРОТОТИПА ИНТЕЛЛЕКТУАЛЬНОЙ СИСТЕМЫ ДЛЯ ОБРАБОТКИ И ВИЗУАЛИЗАЦИИ РЕЗУЛЬТАТОВ РАБОТЫ НЕЙРОННЫХ СЕТЕЙ ПРИ РЕШЕНИИ ЗАДАЧ ДЕТЕКЦИИ, КЛАССИФИКАЦИИ И СЕГМЕНТАЦИИ

Научная статья

Кашинцева О.А.[1,] *, Щербаков А.Е.[2]
[1] ORCID : https://orcid.org/0000-0001-5185-1292;
[2] ORCID : 0009-0000-5883-3230;
[1, 2] Череповецкий государственный университет, Череповец, Российская Федерация

* Корреспондирующий автор (oakashintceva[at]chsu.ru)

Аннотация

В настоящее время большинство известных приложений, которые используют нейронные сети для обработки видео или изображений, разработаны для решения конкретных задач, поэтому имеют свои особенности. Программистам для каждой новой задачи детекции, классификации и сегментации для определенной архитектуры нейронной сети требуется создавать дополнительное программное обеспечение (ПО) с учетом всех зависимостей и особенностей проекта. В статье обосновывается необходимость создания для обработки и визуализации результатов работы нейронных сетей общей универсальной программы, сформулированы критерии к ней. Описывается разработка прототипа такой интеллектуальной системы, который позволит получить требуемый результат сразу, без предварительного создания целого ряда программных модулей, тем самым уменьшая время на разработку продукта.

Ключевые слова: детекция, классификация, сегментация, нейронные сети, интеллектуальная система.

DEVELOPMENT OF A PROTOTYPE OF AN INTELLIGENT SYSTEM FOR PROCESSING AND VISUALIZATION OF NEURAL NETWORK RESULTS WHEN SOLVING DETECTION, CLASSIFICATION AND SEGMENTATION TASKS

Research article

Kashintseva O.A.[1,] *, Shcherbakov A.E.[2]
[1] ORCID : https://orcid.org/0000-0001-5185-1292;
[2] ORCID : 0009-0000-5883-3230;
[1, 2] Cherepovets State University, Cherepovets, Russian Federation

* Corresponding author (oakashintceva[at]chsu.ru)

Abstract

Currently, most of the known applications that use neural networks for video or image processing are designed for specific tasks, so they have their own features. Programmers for each new task of detection, classification and segmentation for a particular neural network architecture need to create additional software (software) taking into account all the dependencies and specifics of the project. The article substantiates the necessity of creating a common universal program for processing and visualization of neural network results, and formulates criteria for it. It describes the development of a prototype of such an intelligent system, which will allow to obtain the required result at once, without preliminary creation of a number of software modules, thus reducing the time for product development.

Keywords: detection, classification, segmentation, neural networks, intelligent system.

Введение

В настоящее время одним из основных трендов в мире является использование нейронных сетей для решения задач детекции, классификации и сегментации во всех областях деятельности человека: медицинской, экономической, научной и др. Например, для обнаружения и классификации опухолей на МРТ снимках, выявления дефектов на дорожных полотнах и на металлоизделиях, для гранулометрии и т.д.

На рынке программного обеспечения имеется множество самых разнообразных моделей нейронных сетей для решения подобных задач [1], [2], [3]. Большинство приложений для детекции, сегментации или классификации объектов на изображении, использующих нейронные сети, имеют общие конструктивные особенности и одинаковые функции. Например, функцию загрузки изображения для обработки или функции для сохранения результатов в базу. Но при этом каждая задача имеет определенные специфичные особенности, в связи с чем и нейронная сеть разработана для решения именно этой конкретной задачи и тоже имеет свои «нюансы». Где-то необходимо перед работой нейронной сети дополнительно изменять изображение (постпроцессинг, препроцессинг, например, привести изображение к определенному разрешению). Какие-то сети требуют специфических сопрограмм (разные сети построены на разных библиотеках). В частности, при работе со снимками МРТ для детекции и классификации опухолей головного мозга необходимо дополнительно декодировать их из архива с записями, потому что требуется принимать данные на вход не в виде изображений, а в виде dicom-архива и в дальнейшем «распаковывать» их и обрабатывать все разом. Или в задачах по определению спелости продуктов необходимо использовать две сети, а не одну: одна – для детекции продукта на изображении, другая – для классификации его спелости/неспелости. В

направлении обнаружения дорожных ям существует потребность соотносить одни и те же объекты на соседних изображениях.

Следовательно, для всякой конкретной задачи требуется создавать свою программную систему с учетом всех особенностей и зависимостей, что является трудоемким процессом, приводящим к затрате временных и экономических ресурсов компаний и разработчиков. Особенно это ощутимо на ранних этапах решения задачи, когда не до конца понятно, решается ли она с помощью нейронных сетей в принципе. Для бизнеса любого размера время создания конечного продукта – это критерий (показатель) конкурентоспособности ИТ-предприятия: чем меньше время, необходимое на разработку продукта, который использует машинное обучение, тем более рентабельным он становится.

На сегодняшний день существует потребность в создании общей универсальной программы для визуализации результатов работы нейронных сетей при решении задач детекции, классификации и сегментации. Такой программный комплекс позволит «оборачивать» различные модели нейронной сети в сервис с одинаковыми методами для взаимодействия с ним, в который можно будет встроить любую модель нейронной сети, и он автоматически предоставит пользовательский интерфейс и набор методов для взаимодействия с другими программными системами без написания целого ряда модулей.

Целью данного исследования является разработка критериев к программной системе для обработки изображений нейронными сетями и дальнейшей визуализации этих результатов при решении задач детекции, классификации и сегментации.

Для достижения цели необходимо решить следующие задачи:

1. Проанализировать предметную область.
2. Рассмотреть возможные форматы для реализации обработки изображений сетями, архитектуры программного комплекса и программные технологии (языки программирования, фреймворки, способы контейнеризации).
3. Разработать прототип серверного программного обеспечения с возможностью доступа по API и пользовательский интерфейс самой системы.
4. Протестировать прототип в нейросетевых проектах.
5. Сформулировать критерии к программной системе.

Методы и принципы исследования

Для того чтобы сформировать основные критерии к общей универсальной программной системе для обработки и визуализации результатов работы нейронных сетей при решении задач детекции, классификации и сегментации и разработать прототип такой системы был проведен анализ литературных источников и существующих программных решений [4], [5], [6], [7]. Анализ показал, что важными моментами при реализации подобной системы являются время инференса и необходимое для запуска модели окружение.

Время инференса может быть как почти мгновенным, так и занимать от нескольких секунд до нескольких десятков секунд в зависимости от сложности конкретной модели. Из-за этого при проектировании системы следует учесть возможную задержку между отправкой изображения на обработку и получением результата работы нейронной сети [8].

Для запуска нестандартных в рамках системы моделей требуется обеспечить возможность подключения подходящего окружения. Лучшим вариантом для этого будет упаковка обработчика данной модели и ее зависимостей в отдельный контейнер, с возможностью обращения к нему при помощи REST API или подключения данного контейнера к брокеру сообщений [9].

Рассмотрев различные программные решения, был сделан вывод о компонентах, из которых может состоять общая структура универсальной системы:

· ядро системы (API Gateway);
· очередь сообщений (Redis);
· система управления обработчиками (Celery);
· система агрегации логов (Loki);
· пользовательский интерфейс (UI).

Для удобства развертывания и дальнейшего использования системы следует собрать ее в несколько docker-контейнеров или в единый docker-compose.yml.

Чтобы обеспечить универсальность системы, было решено использовать формат ONNX, один из самых популярных и активно развивающихся форматов [8], [10]. Он предоставляет интерфейс для взаимодействия с моделью, конвертируя веса различных нейросетевых моделей в .onnx файлы.

Для сохранения возможности использования нестандартных моделей следует реализовать абстрактный класс обработчика, на основе которого может быть создан обработчик для какой-либо модели в отличном от ONNX формате.

Разрабатываемая программная система должна быть расширяемой в связи с тем, что задачи, решаемые с помощью нейронных сетей, могут иметь различные подзадачи. Решить данную проблему можно с помощью модульного подхода, где каждая подзадача изолируется в своем модуле.

Для удобства проверки всех основных функций системы следует реализовать простейший графический интерфейс.

Исходя из анализа существующих решений, для создания прототипа были выбраны следующие программные технологии:

· Python – основной язык программирования для разработки backend-части,
· JavaScript – язык программирования для создания графического интерфейса системы,
· Vue.js – фреймворк для создания графического интерфейса,
· Webpack – фреймворк для упаковки графического интерфейса,

· Docker – система контейнеризации.

Разработка прототипа

Всю проектируемую систему можно условно разделить на две больших части: «Фронтенд» и «Бэкэнд». В «Фронтенд» (внешняя часть программы) входит пользовательский интерфейс и прочие компоненты, с которыми непосредственно взаимодействует пользователь. «Бэкэнд» – это часть программы, которая непосредственно работает с данными: получает их, сохраняет и обрабатывает при помощи нейронной сети. Она должна удовлетворять нескольким важным требованиям:

· Отказоустойчивость. При неработоспособности системы или ее отдельных компонентов система должна быть в состоянии сообщить «Фронтенду» об этом происшествии.

· Расширяемость. При необходимости должна быть возможность расширить систему для увеличения ее пропускной способности.

· Производительность. Система должна обрабатывать запросы от пользователей максимально быстро и эффективно.

· Совместимость. «Бэкэнд» должен быть совместим с широким диапазоном клиентских приложений и устройств.

Для реализации этих требований в программе отдельные ее модули были спроектированы максимально независимо и расширяемо в виде отдельных сервисов. Структурная схема программы представлена на рисунке 1 (См. рисунок 1).

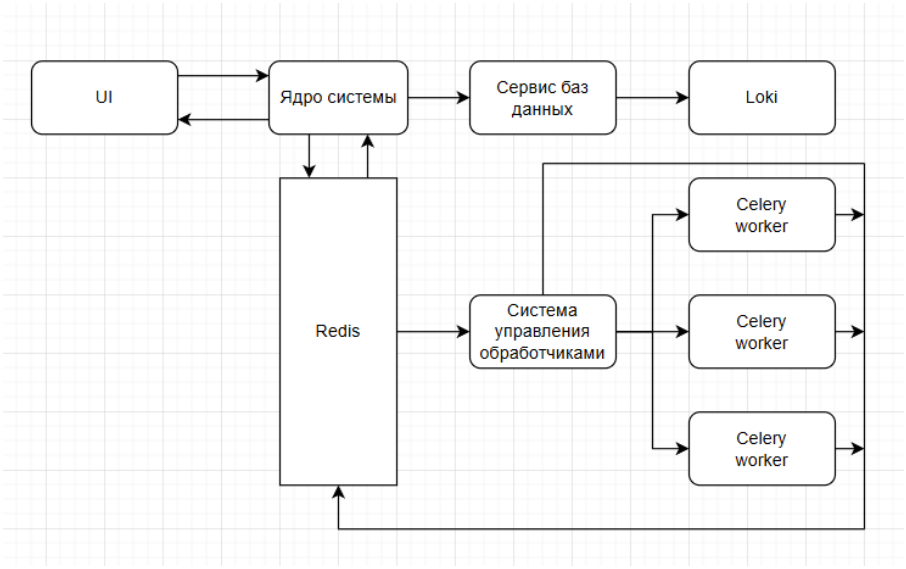

Рисунок 1 - Структурная схема системы

DOI: https://doi.org/10.60797/IRJ.2024.145.120.1

Ядро Системы – сервис, который отвечает за управление и передачу данных внутри всей системы, является входной точкой при получении данных от UI (внешняя часть программы, с которой взаимодействует пользователь), обеспечивает совместимость системы через интерфейс REST API.

Система управления обработчиками – основной сервис, который непосредственно обеспечивает обработку изображения при помощи нейронной сети. Инкапсулирует в себе всю работу с нейросетевыми моделями и всю логику инференса. Реализован на основе абстрактного класса и имеет два метода: *run* и *run_pack*. Метод *run* служит для обработки одного изображения, а *run_pack* позволяет обработать серию изображений. При получении нового запроса из очереди сообщений сервис управления обработчиками запускает задание в одном из нескольких celery-worker’ов, которые в свою очередь запускают процесс инференса. Данные worker’ы могут быть расположены на разных физических серверах и, соответственно, могут использовать дополнительные аппаратные мощности. Система управления обработчиками при помощи *healthcheck’ов* проверяет статус worker’ов, и, в случае их недоступности, сигнализирует об этом ядру системы, что позволяет корректировать нагрузку на систему или не отправлять данные, если система недоступна в целом. В систему управления обработчиками могут быть добавлены различные конфигурации для celery-worker’ов, что предоставляет возможность использовать различные модели нейронных сетей в одном приложении с единым интерфейсом.

Сервис Базы Данных реализует всю работу с СУБД и служит для сохранения входных данных и результатов обработки нейронными сетями. У пользователя всегда есть возможность запросить историю обработок и результаты предыдущих обработанных изображений.

С помощью открытой системы для агрегации, поиска и хранения журналов (логов) микросервисов Loki можно эффективно управлять логами в больших масштабах, просматривать и анализировать журналы разных сервисов в едином интерфейсе.

Требования к внешней части программы («Фронтенд») следующие:

· Адаптивность – интерфейс должен быть доступен с различных устройств.

· Кроссплатформеность – интерфейс должен быть доступен в различных операционных системах, без дополнительных изменений внутри приложения.

С учетом требований пользовательский интерфейс прототипа реализован в виде веб-приложения. В связи с этим языком программирования для разработки стал JavaScript, языком гипертекстовой разметки – HTML, языком описания стилей – CSS. Помимо базового JavaScript использовался один из реактивных фреймворков данного языка программирования VueJS из-за его простоты и мощного функционала. Для внутренней части программы применялся язык Python с асинхронным фреймворком FastAPI, в котором удобно создавать REST API приложение. Для взаимодействия с базами данных используется библиотека «SQLAlchemy» для языка Python вместе с СУБД PostgresSQL, что позволяет избежать возможных уязвимостей, связанных с языком SQL.

Тестирование прототипа

Прототип системы был протестирован на различных обученных моделях нейронных сетей, созданных для

1) детекции и сегментации опухолей головного мозга на снимках МРТ,

2) сегментации гранулометрического состава,

3) детекции и классификации номеров на трубах. В качестве метрик были выбраны время обработки изображения и потребление CPU и RAM.

1. Тестирование на модели детекции и сегментации опухолей головного мозга на снимках МРТ.

Запуск нейросетевого сервиса проводился с использованием CPU. Среднее время обработки одного изображения – 6 секунд. Обработка без использования системы заняла в среднем 5 секунд. Статистика ресурсов показывает малое потребление CPU и RAM, так как на изображении присутствует малое количество объектов, которые нужно детектировать. Результаты работы ПО представлены на рисунке 2 (см. рисунок 2*a*).

2. Тестирование на модели сегментации гранулометрического состава.

Запуск нейросетевого сервиса проводился с использованием CPU. Среднее время обработки одного изображения – 24 секунды. Обработка без использования системы заняла в среднем 21 секунду. Статистика ресурсов показывает высокое потребление CPU и RAM, так как на изображении присутствует большое количество объектов, которые нужно детектировать. Результаты работы ПО представлены на рисунке 2 (см. рисунок 2*b*).

3. Тестирование на модели детекции и классификации номеров на трубах.

Запуск нейросетевого сервиса проводился с использованием CPU. Среднее время обработки одного изображения – 3 секунды. Обработка без использования системы заняла в среднем 3 секунды. Статистика ресурсов показывает малое потребление CPU и RAM, так как на изображении присутствует малое количество объектов, которые нужно детектировать. В данной задаче используется две сети (одна находит номер на фото, вторая его распознает). Низкое потребление ресурсов представляет разработанное ПО с лучшей стороны. Результаты работы ПО представлены на рисунке 2 (см. рисунок 2*c*).

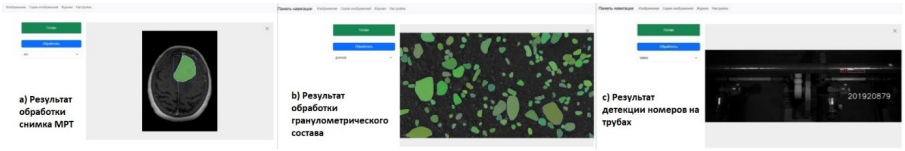

Рисунок 2 - Результаты тестирования

DOI: https://doi.org/10.60797/IRJ.2024.145.120.2

Тестирование показало, что разработанное ПО выполнило задачу за приемлемое время и без ошибок.

Основные результаты

В результате исследования были предложены следующие критерии к универсальной программной системе для обработки и визуализации результатов работы нейронных сетей при решении задач детекции, классификации и сегментации:

1) приемлемое время инференса;

2) параллельное выполнение задач;

3) изоляция моделей;

4) расширяемость системы;

5) использование универсальных форматов нейронных сетей.

С учетом данных критериев были выбраны технологии, и на их основе создан прототип интеллектуальной системы. Разработанное ПО было протестировано в проектах детекции опухолей на снимках МРТ, дефектов на металлоизделиях, номеров на трубах, где показало свою работоспособность: прототип справился со всеми задачами за приемлемое время.

Направление для дальнейшей работы: создание полноценной системы визуализации и обработки результатов работы нейронных сетей, решение возможных проблем с производительностью и масштабируемостью, добавление совместимости с различными дополнительными моделями нейронных сетей.

Заключение

Разработанный на основе предложенных критериев прототип программной системы позволит не только ускорить разработку новых решений в сфере использования нейронных сетей и машинного зрения, но и существенно сократить время на создание минимально жизнеспособного конечного продукта, упростить прототипирование новых идей и

повысить безопасность разработки новых систем на ранних этапах (уменьшить количество потенциальных уязвимостей).

Конфликт интересов

Не указан.

Рецензия

Все статьи проходят рецензирование. Но рецензент или автор статьи предпочли не публиковать рецензию к этой статье в открытом доступе. Рецензия может быть предоставлена компетентным органам по запросу.

Conflict of Interest

None declared.

Review

All articles are peer-reviewed. But the reviewer or the author of the article chose not to publish a review of this article in the public domain. The review can be provided to the competent authorities upon request.

Список литературы / References

1. 30 нейросетей для работы с изображениями и видео. — URL: https://tproger.ru/articles/neural-img-and-video/ (дата обращения 15.04.2024)
2. Deep Learning. — URL: https://www.nature.com/articles/nature14539 (accessed: 10.04.2024)
3. Горяшин К.С. Обработка изображений и машинное зрение / К.С. Горяшин, И.А. Толстухин. — СПб.: Питер, 2019. — 384 с.
4. Ньюмен С. Создание микросервисов / С. Ньюмен. — 2-е изд. — СПб.: Питер, 2023. — 624 с.: ил. — (Серия «Бестселлеры O’Reilly»).
5. Матвеенко С.Н. Основы глубокого обучения: Курс лекций / С.Н. Матвеенко, В.А. Хроленко. — Уфа: Издательство Уфимского университета – 2019. — 94 с.
6. Голубцов К.Ю. Python и машинное обучение: современный практический подход к разработке приложений / К.Ю. Голубцов, Ю.П. Бондаренко. — М.: ДМК Пресс, 2019. — 416 с.
7. Седунов А.Ю. Разработка и исследование нейронных сетей в приложениях / А.Ю. Седунов, Л.В. Сорокин. — М.: ДМК Пресс, 2018. — 256 с.
8. Документация celery. — URL: https://docs.celeryq.dev/en/stable/ (дата обращения: 15.04.2024)
9. Документация docker. — URL: https://docs.docker.com/ (дата обращения: 15.04.2024)
10. Документация ONNX. — URL: https://onnx.ai/ (дата обращения: 15.04.2024)

Список литературы на английском языке / References in English

1. 30 nejrosetej dlya raboty s izobrazheniyami i video [30 neural networks for working with images and videos]. — URL: https://tproger.ru/articles/neural-img-and-video/ (accessed: 15.04.2024) [in Russian]
2. Deep Learning. — URL: https://www.nature.com/articles/nature14539 (accessed: 10.04.2024)
3. Goryashin K.S. Obrabotka izobrazhenij i mashinnoe zrenie [Image processing and machine vision] / K.S. Goryashin, I.A. Tolstuhin. — SPb.: Piter, 2019. — 384 p. [in Russian]
4. N'yumen S. Sozdanie mikroservisov [Creating microservices] / S. N'yumen. — 2nd ed. — SPb.: Piter, 2023. — 624 p.: il. — (The O'Reilly Bestseller Series) [in Russian].
5. Matveenko S.N. Osnovy glubokogo obucheniya: Kurs lekcij [The basics of deep learning: A course of lectures] / S.N. Matveenko, V.A. Hrolenko. — Ufa: Ufa University Press – 2019. — 94 p. [in Russian]
6. Golubcov K.YU. Python i mashinnoe obuchenie: sovremennyj prakticheskij podhod k razrabotke prilozhenij [Python and Machine Learning: a modern practical approach to application development] / K.YU. Golubcov, YU.P. Bondarenko. — M.: DMK Press, 2019. — 416 p. [in Russian]
7. Sedunov A.YU. Razrabotka i issledovanie nejronnyh setej v prilozheniyah [Development and research of neural networks in applications] / A.YU. Sedunov, L.V. Sorokin. — M.: DMK Press, 2018. — 256 p. [in Russian]
8. Dokumentaciya celery [Documentation sellers]. — URL: https://docs.celeryq.dev/en/stable/ (accessed: 15.04.2024) [in Russian]
9. Dokumentaciya docker [Docker documentation]. — URL: https://docs.docker.com/ (accessed: 15.04.2024) [in Russian]
10. Dokumentaciya ONNX [Documentation ONNX]. — URL: https://onnx.ai/ (accessed: 15.04.2024) [in Russian]