

Д. А. Стригун^{1}, П. Ю. Бугаков¹*

Разработка приложения для выявления брака в коллекции изображений с использованием нейросетевых технологий

¹ Сибирский государственный университет геосистем и технологий, г. Новосибирск,
Российская Федерация
* e-mail: linkeert@yandex.ru

Аннотация. Статья посвящена разработке приложения, способного выявлять брак в коллекции изображений с использованием нейросетевых технологий. Целью работы является разработка веб-приложения, использующего модель машинного обучения на основе сверточных нейронных сетей для распознавания и классификации бракованных коллекций изображений. Основной задачей является определение «качественных» и «некачественных» изображений из большого объема данных, получаемых из различных источников, перед их использованием для планирования и анализа. В рамках данной работы была выбрана архитектура VGG в качестве основы для сверточной нейронной сети. Приложение построено на клиент-серверной архитектуре. В статье приводятся основные типы брака в изображениях, а также причины их возникновения, описаны существующие методы выявления брака, включая традиционные алгоритмы обработки изображений и современные нейросетевые подходы, с акцентом на СНС. В практической части работы описаны результаты реализации приложения, его тестирования на реальных данных.

Ключевые слова: веб-приложение, качество изображений, геоданные

D. A. Strigun^{1}, P. Yu. Bugakov¹*

Development of an Application for Detecting Defects in a Collection of Images Using Neural Network Technologies

¹ Siberian State University of Geosystems and Technologies, Novosibirsk, Russian Federation
* e-mail: linkeert@yandex.ru

Abstract. The article is devoted to the development of an application capable of detecting defects in a collection of images using neural network technologies. The aim of the work is to develop a web application using a machine learning model based on convolutional neural networks for recognizing and classifying defective image collections. The main task is to identify "high-quality" and "low-quality" images from a large amount of data obtained from various sources before using them for planning and analysis. As part of this work, the VGG architecture was chosen as the basis for a convolutional neural network. The application is built on a client-server architecture. The article presents the main types of marriage in images, as well as the causes of their occurrence, describes existing methods for detecting marriage, including traditional image processing algorithms and modern neural network approaches, with an emphasis on SNA. In the practical part of the work, the results of the application implementation and its testing on real data are described.

Keywords: web application, image quality, geodata

Введение

Актуальность данной работы обусловлена растущей потребностью в автоматизации процесса выявления брака в коллекциях изображений, полученных автоматизированными способами при проведении автоматизированной назем-

ной и аэрофотосъемки. Традиционные методы, основанные на ручном анализе или простых алгоритмах обработки изображений, зачастую не справляются с этой задачей из-за своей трудоемкости, субъективности и ограниченной эффективности. Применение сверточных нейронных сетей (СНС) для выявления брака в изображениях открывает новые возможности для повышения точности, скорости и объективности анализа, что в конечном итоге способствует принятию более обоснованных решений на основе геоданных.

Цель работы заключается в разработке веб-приложения, использующего модель машинного обучения на основе СНС для распознавания и классификации бракованных изображений в коллекциях.

Для достижения цели должны быть решены следующие задачи:

- выполнить анализ предметной области и определить основные типы брака в изображениях, а также причины их возникновения;
- исследовать существующие методы выявления брака, включая традиционные алгоритмы обработки изображений и современные нейросетевые подходы, с акцентом на СНС;
- выбрать оптимальную архитектуру СНС и разработать модель обучения для выявления брака в изображениях;
- реализовать приложение, интегрировав в него обученную модель СНС, и провести тестирование его работы на реальных данных.

Практическая ценность данной работы заключается в создании инструмента, который позволит автоматизировать и оптимизировать процесс анализа геоданных, что приведет к повышению эффективности работы специалистов в различных областях.

Методы и материалы

Обнаружение дефектов изображения является важной задачей в области обработки изображений и компьютерного зрения [1]. Существует множество способов решения этой проблемы, которые можно разделить на две основные категории: традиционные алгоритмы обработки изображений и современные методы, основанные на нейронных сетях.

К традиционным алгоритмам обработки изображений можно отнести фильтр – применяется для сглаживания, повышения резкости или выделения определенных элементов изображения, а также преобразование – используется для анализа изображения каждой области (частота, вейвлет).

Современными методами основанных на нейронных сетях можно считать классификацию и сегментацию [2].

Классификация заключается в том, что нейронная сеть обучается классифицировать изображения как «качественные» или «некачественные». С этой целью используется сверточная нейронная сеть, которая может извлекать сложные признаки из изображения и принимать решения на основе этих признаков [3–6]. При выполнении сегментации нейронные сети обучаются распознавать области изображения, содержащие дефекты. Это позволяет не только определить наличие брака, но и локализовать их на изображении.

В разработке приложения использованы языки программирования Python для реализации серверной части приложения и JavaScript для создания клиентской части. Основными фреймворками и библиотеками для разработки приложения способного выявлять брака в коллекции изображений с использованием нейросетевых технологий можно считать:

- Flask. Используется для создания веб-приложения [7];
- TensorFlow. В качестве библиотеки для проектирования и обучения модели нейросети [8];
- NumPy. Для работы с массивами [9];
- OpenCV. Для обработки изображений и видеол [1].

Точность классификации, скорость обработки, простота использования, надежность и расширяемость – это основные показатели успешной реализации приложения.

В рамках данной работы была выбрана архитектура VGG в качестве основы для сверточной нейронной сети. VGG представляет собой глубокую нейронную сеть, состоящую из последовательности сверточных слоев, слоев пулинга и полносвязных слоев.

Результаты

Качественные изображения были получены с ресурсов Pixabay и Pexels через их API, что позволило автоматизировать сбор данных и обеспечить разнообразие изображений.

Для создания набора данных с различными типами дефектов, изображения были преднамеренно испорчены следующими методами:

1. Гауссово размытие. Для имитации размытия движения и фокуса.
2. Добавление случайного шума с разной интенсивностью.
3. Затемнение с использованием фильтра enhance из библиотеки PIL для эффекта недоэкспонирования.
4. Пересветы с помощью коррекции гаммы с использованием OpenCV. Для имитации переэкспонирования.

Приложение состоит из клиентской и серверной частей. Клиентская часть – то, что видит и с чем взаимодействует пользователь. Серверная часть – «мозг и сердце» приложения. Она обрабатывает запросы клиента, взаимодействует с моделью нейросети, формирует и отправляет результаты обратно клиенту.

Взаимодействие этих компонентов представлено на диаграммах последовательности (рис. 1).

Интерфейс реализован с использованием следующих технологий: html, css, javascript. Результат реализации интерфейса приложения представлен на рис. 2.

После успешного обучения модели VGG на подготовленном наборе данных, следующим этапом стало ее интегрирование в серверную часть приложения [10, 11]. Этот процесс обеспечил связующее звено между пользовательским интерфейсом и аналитическим потенциалом нейронной сети.

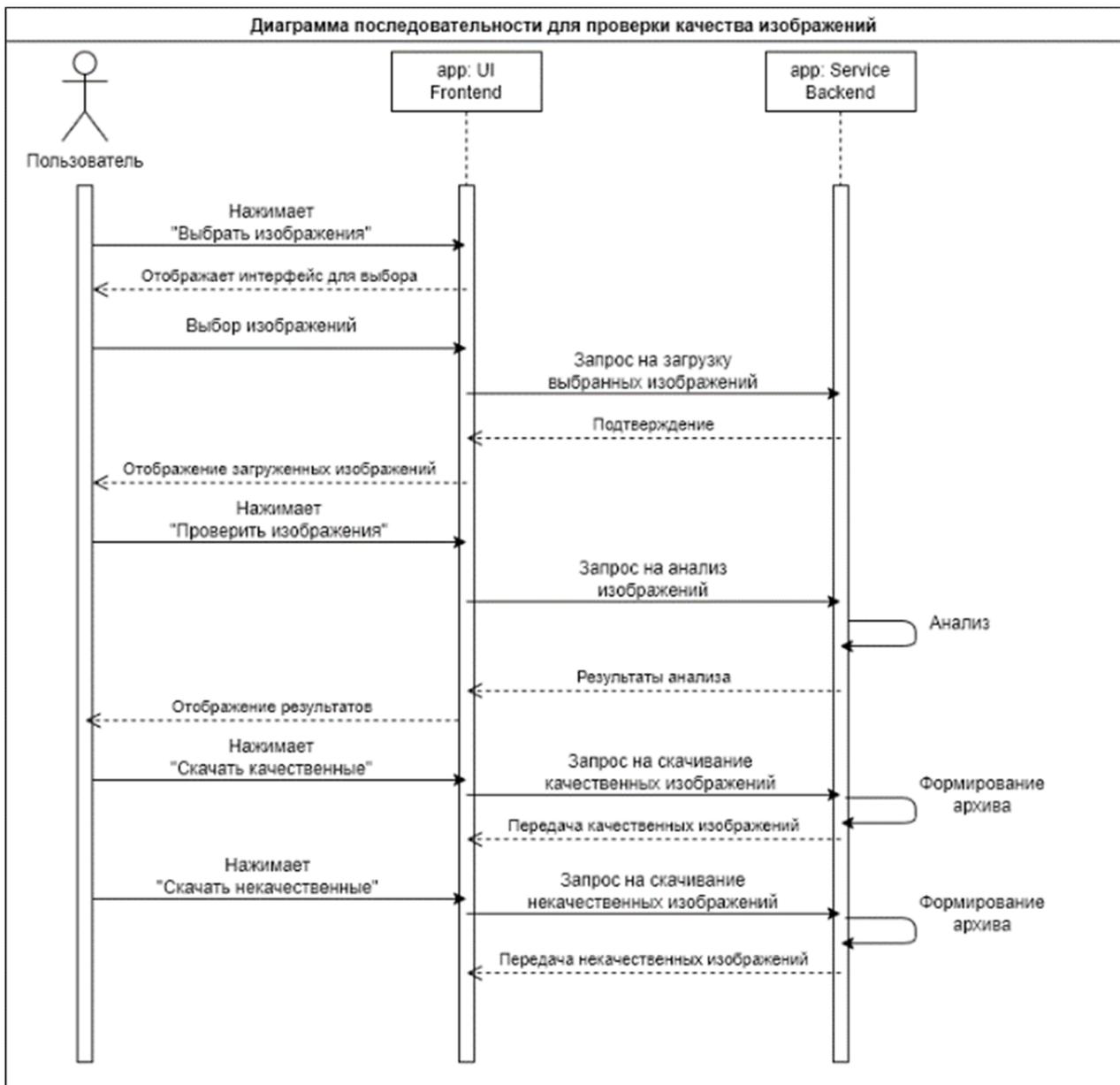


Рис. 1. Диаграмма последовательности, страница «Проверка качества изображений»

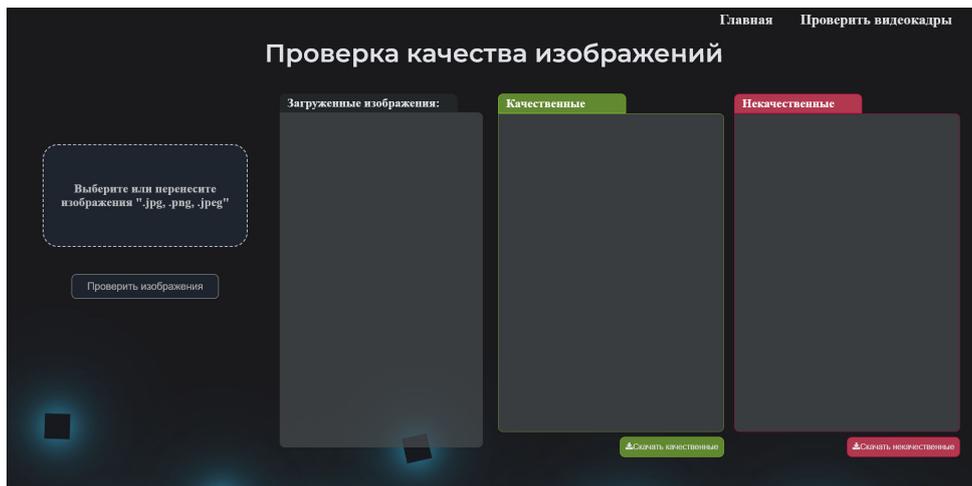


Рис. 2. Интерфейс приложения, страница «Проверка качества изображений»

При получении запроса от пользователя, содержащего изображения или видео, серверная часть приложения инициирует цепочку действий. Сначала происходит декодирование полученных данных: изображения и видеокadres преобразуются из форматов JPEG, PNG или MP4 в массивы NumPy, понятные для нейронной сети. Затем выполняется предобработка данных, включающая нормализацию и изменение размера, чтобы привести их к виду, который использовался при обучении модели.

На стороне клиента, технология AJAX обеспечивает асинхронную коммуникацию с сервером, позволяя обновлять результаты анализа в режиме реального времени без перезагрузки страницы. Полученные от сервера результаты отображаются в соответствующих разделах интерфейса («Качественные» или «Некачественные»), а пользователь имеет возможность скачать архивы с отсортированными изображениями или видеокadresами.

Основным показателем производительности приложения является среднее время обработки одного изображения. В ходе тестирования, среднее время обработки составило 0,2 секунды (рис. 3).

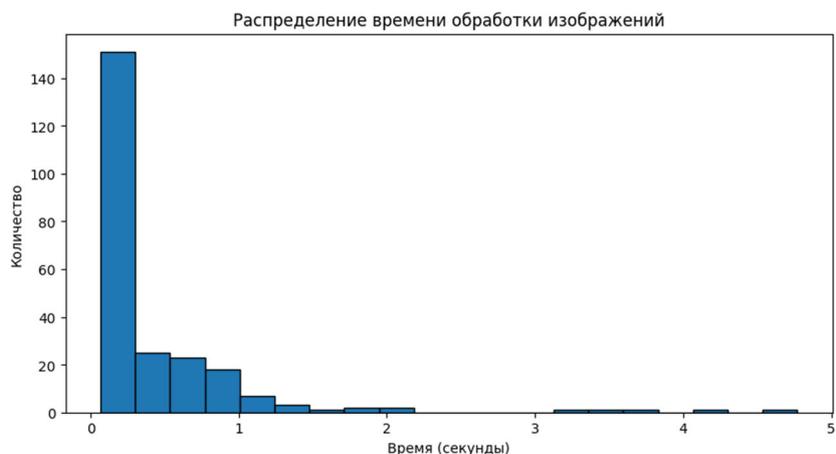


Рис. 3. Гистограмма распределения времени обработки изображений

Для оценки качества работы приложения были использованы стандартные метрики классификации: accuracy (точность), precision (точность), recall (полнота), f1-score. Полученные результаты метрик наглядно представлены на рис. 4.

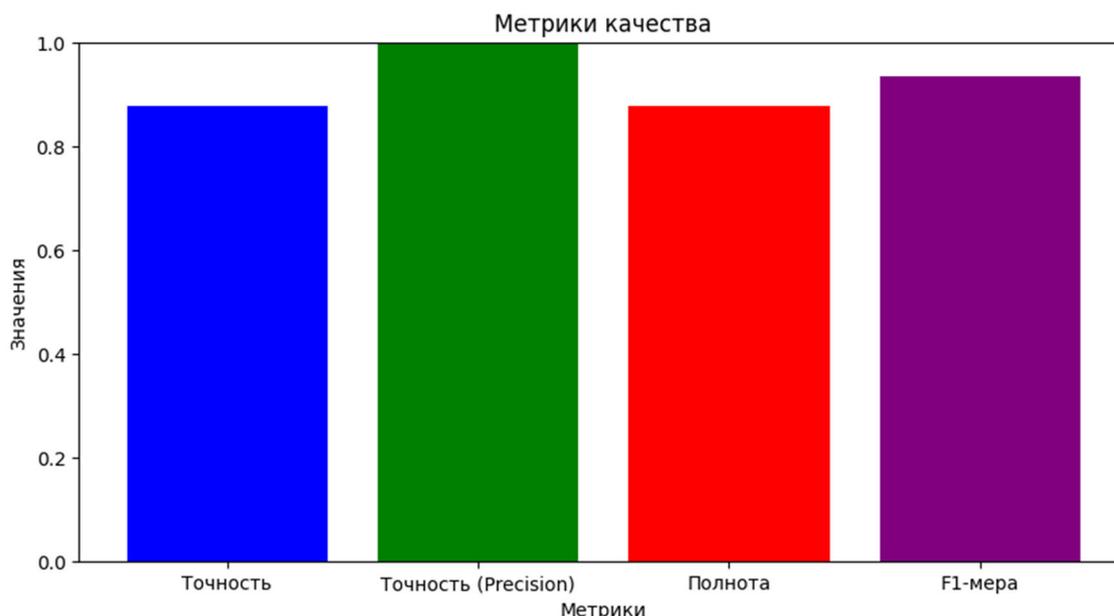


Рис.4. График полученных результатов метрик

Заключение

В рамках данной работы было разработано веб-приложение, которое предназначено для выявления брака в коллекции изображений с использованием нейросетевых технологий.

Приложение реализовано на основе клиент-серверной архитектуры, используя язык программирования Python и фреймворк Flask.

Практическая значимость данного проекта заключается в возможности существенной оптимизации процесса контроля качества аэрофотосъемки. Автоматическое выявление бракованных снимков позволяет сократить временные и финансовые затраты, а также повысить объективность и точность оценки качества данных. Это, в свою очередь, способствует повышению качества картографических продуктов, эффективности геологических исследований, оптимизации сельскохозяйственного производства и принятию более обоснованных решений в других областях, использующих аэрофотосъемку.

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1 Освоение OpenCV с помощью Python: Полное руководство по обработке изображений и компьютерному зрению // Machine learning: – 2023. – URL: <https://vc.ru/u/1389654-machine-learning/661520-osvoenie-opencv-s-pomoshyu-python-polnoe-rukovodstvo-po-obrabotke-izobrazhenii-i-kompyuternomu-zreniyu> (дата обращения: 21.01.2024).

2 Орешков В. Классификация данных при помощи нейронных сетей – 2021 – URL: <https://loginom.ru/blog/neural-classification> (дата обращения: 19.01.2024).

- 3 Подробное руководство по свёрточным нейронным сетям // Наука о данных: – 2020. – URL: <https://nuancesprog.ru/p/8307/> (дата обращения: 16.01.2024).
- 4 Сверточная нейронная сеть (CNN): [сайт]. – URL: <https://www.tensorflow.org/tutorials/images/cnn?hl=ru> (дата обращения: 18.01.2024). – Текст: электронный.
- 5 Бычков А.Г., Киселёва Т.В., Маслова Е.В. Использование сверточных нейросетей для классификации изображений // Вестник Сибирского государственного индустриального университета. 2023. №1 (43). С. 39-49.
- 6 Сикорский, О.С. Обзор свёрточных нейронных сетей для задачи классификации изображений / О. С. Сикорский // Новые информационные технологии в автоматизированных системах – Москва, 2017. – № 20. – СС. 37-42.
- 7 Flask Руководство пользователя: [сайт]. – URL: <https://flask.palletsprojects.com/en/2.2.x/> (дата обращения: 12.02.2024). – Текст: электронный.
- 8 TensorFlow Руководство: [сайт]. – URL: <https://www.tensorflow.org/tutorials?hl=ru> (дата обращения: 14.01.2024). – Текст: электронный.
- 9 Обработка изображений с помощью Python, NumPy // Обработка изображений: – 2020. – URL: <https://note.nkmk.me/en/python-numpy-image-processing/> (дата обращения: 22.01.2024).
- 10 Keras: [сайт]. – URL: <https://blog.skillfactory.ru/glossary/keras/> (дата обращения: 14.01.2024). – Текст: электронный.
- 11 Как подготовить набор изображений для обучения нейронной сети в Keras: [сайт]. – URL: https://master--hardcore-noether-529719.netlify.app/deep_learning/2018/01/06/How-to-Prepare-Image-Dataset-for-Keras.html (дата обращения: 19.01.2024). – Текст: электронный.

© Д. А. Стригун, П. Ю. Бугаков, 2024