

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

Кафедра математической кибернетики и компьютерных наук

**РАЗРАБОТКА ВЕБ-ПРИЛОЖЕНИЯ ДЛЯ СТИЛИЗАЦИИ
ИЗОБРАЖЕНИЯ С ИСПОЛЬЗОВАНИЕМ НЕЙРОННОГО ПЕРЕНОСА
СТИЛЯ НА ОСНОВЕ СВЕРТОЧНОЙ СЕТИ VGG19**

АВТОРЕФЕРАТ БАКАЛАВРСКОЙ РАБОТЫ

студента 4 курса 411 группы
направления 02.03.02 — Фундаментальная информатика и информационные
технологии
факультета КНиИТ
Бахтина Егора Олеговича

Научный руководитель
доцент., к. ф.-м. н.

Ю. Н. Кондратова

Заведующий кафедрой
доцент, к. ф.-м. н.

С. В. Миронов

Саратов 2025

ВВЕДЕНИЕ

В современном мире визуальная информация играет важную роль в коммуникации и восприятии окружающего мира. С развитием цифровых технологий и распространением интернета люди всё чаще используют изображения и фотографии для обмена эмоциями, идеями и впечатлениями. Визуальный контент стал неотъемлемой частью социальных сетей, блогов, новостных порталов и многих других онлайн-платформ. Вместе с этим вырос интерес к инструментам, позволяющим обрабатывать и изменять изображения, делая их более выразительными и привлекательными.

Одним из перспективных направлений обработки изображений является стилизация, то есть преобразование фотографии или рисунка в стиле другого изображения, например, знаменитой картины или определённого художественного жанра. Данная технология позволяет создавать уникальные художественные работы, объединяя содержание одного изображения и стиль другого. Особое место в этой области занимает метод нейронного переноса стиля (Neural Style Transfer), основанный на использовании глубоких сверточных нейронных сетей. Суть метода заключается в выделении и разделении контентной и стилевой информации с помощью обученной модели, после чего происходит создание нового изображения, сочетающего эти два компонента.

Широкое распространение получили модели, построенные на основе архитектуры VGG19 — одной из самых популярных и эффективных сверточных нейронных сетей, применяемых для извлечения признаков изображений. Использование таких моделей позволяет получать качественные результаты стилизации при относительно умеренных затратах вычислительных ресурсов.

Веб-приложения, использующие нейронный перенос стиля, представляют собой удобный инструмент для пользователей, позволяющий быстро и просто преобразовывать изображения, не требуя специальных знаний в области программирования или машинного обучения. Однако разработка подобных приложений требует комплексного подхода, включающего выбор и интеграцию моделей искусственного интеллекта, создание интуитивно понятного пользовательского интерфейса и обеспечение стабильной работы на серверной части.

Актуальность данного проекта обусловлена растущей популярностью инструментов для творчества с использованием искусственного интеллекта, а также востребованностью веб-решений, доступных с различных устройств и плат-

форм. Создание эффективного веб-приложения для стилизации изображений на основе нейронного переноса стиля позволит расширить возможности пользователей и повысить доступность современных технологий в области компьютерного зрения.

Целью данной работы является разработка веб-приложения, реализующего нейронный перенос стиля с помощью сверточной нейронной сети VGG19, предоставляющего пользователю удобный интерфейс для загрузки и стилизации изображений.

Для достижения поставленной цели необходимо выполнить следующие задачи:

- изучить теоретические основы нейронного переноса стиля и особенности архитектуры VGG19;
- выбрать и адаптировать модель для выполнения задачи стилизации изображений;
- разработать фронтенд веб-приложения для загрузки и отображения изображений;
- реализовать серверную часть, обеспечивающую обработку изображений и выполнение стилизации;

Структура и объём работы. Бакалаврская работа состоит из введения, 2 глав, заключения, списка использованных источников и приложений. Общий объём работы — 60 страниц, из них 40 страниц — основное содержание, включая 4 рисунка, список использованных источников информации — 20 наименований.

КРАТКОЕ СОДЕРЖАНИЕ РАБОТЫ

Первый раздел «Анализ предметной области» В первом разделе работы проводится всесторонний анализ теоретических основ нейронного переноса стиля, который лежит в основе предлагаемого веб-приложения для стилизации изображений. Вначале описывается постановка задачи нейронного переноса стиля, которая заключается в создании нового изображения, сочетая содержание одного изображения с художественным стилем другого. Этот процесс, основывающийся на методах глубокого обучения, позволяет создавать уникальные визуальные произведения искусства, объединяя два изображения в одном.

Затем следует обзор истории развития методов стилизации изображений, начиная от более традиционных подходов до появления новых методов, основанных на нейронных сетях. В этом контексте рассматриваются первые попытки автоматической стилизации изображений с использованием марковских случайных полей и алгоритмов текстурирования, а также ограниченность этих методов в разделении содержимого и стиля изображения. Далее описывается революция, произошедшая в 2015 году с появлением метода нейронного переноса стиля, предложенного Леоном Гатисом и его коллегами. Этот метод использует глубокие сверточные нейронные сети для разделения содержания и стиля изображения, что позволило значительно улучшить качество стилизации.

Одним из ключевых аспектов работы является использование сверточных нейронных сетей (CNN), которые являются основой для извлечения признаков содержания и стиля изображений. В разделе подробно рассматривается архитектура сверточной нейронной сети VGG19, которая является одной из самых популярных для решения задач компьютерного зрения. Описание включает в себя подробности о каждом из слоев сети, их роли в извлечении признаков различного уровня: от простых признаков, таких как края и текстуры, до более сложных объектов и структур. Также приводится математическое описание работы сверточных слоев, операций подвыборки (max pooling) и активирующих функций, таких как ReLU, которые помогают модели извлекать нелинейные зависимости.

Особое внимание уделяется математической формулировке задачи нейронного переноса стиля, что позволяет более точно понять принципы работы модели. Рассматриваются функции потерь, используемые для минимизации разницы между содержанием и стилем: функция потерь по содержанию ($L_{content}$),

которая измеряет различие между активациями слоев содержания для оригинального и генерируемого изображений, и функция потерь по стилю (L_{style}), которая рассчитывает разницу между стилизованными признаками, извлеченными через матрицы Грама. Описан процесс оптимизации, который сводится к минимизации комбинированной функции потерь, включающей как содержание, так и стиль изображения.

Далее в разделе рассматриваются альтернативные подходы к переносу стиля, которые позволяют улучшить производительность и скорость стилизации изображений. В частности, рассматривается метод Fast Neural Style Transfer, предложенный Джастином Джонсоном и его коллегами, который предлагает обученную сеть, способную за один проход выполнить перенос стиля на изображение. Этот метод значительно быстрее классического подхода, однако ограничен возможностью работать только с одним стилем, что делает его менее гибким. Также анализируется метод AdaIN (Adaptive Instance Normalization), который позволяет производить перенос стиля в реальном времени и поддерживает произвольные стили без необходимости обучения для каждого конкретного стиля. Это открывает возможности для создания более гибких и быстрых приложений, однако требует более сложных архитектур и оптимизации.

После этого раздел переходит к выбору инструментов и технологий для разработки веб-приложения. Рассматривается выбор фронтенд-технологий, таких как HTML, CSS и JavaScript, которые позволяют создать удобный и интуитивно понятный интерфейс для взаимодействия с пользователем. В частности, описываются возможности HTML5 для работы с изображениями в браузере, а также использование CSS3 для создания адаптивного дизайна. JavaScript используется для обработки взаимодействия пользователя с приложением, включая загрузку изображений и асинхронные запросы к серверу для выполнения переноса стиля.

Для реализации серверной части работы выбран язык Python и фреймворк Flask, которые идеально подходят для быстрого создания веб-приложений и интеграции с библиотеками для машинного обучения. Рассматриваются возможности TensorFlow и Keras, которые обеспечивают доступ к предобученным моделям, таким как VGG19, для извлечения признаков из изображений и выполнения нейронного переноса стиля. Использование этих технологий позволяет значительно упростить разработку и ускорить процесс обучения и применения

модели.

Второй раздел «Проектирование и разработка веб-приложения» Во втором разделе работы подробно рассматривается процесс проектирования и разработки веб-приложения для нейронного переноса стиля, использующего сверточную нейронную сеть VGG19. Веб-приложение должно обеспечивать пользователю удобный и интуитивно понятный интерфейс для загрузки изображений, выбора стиля и получения результатов стилизации. В этом разделе подробно описаны архитектурные особенности приложения, включая как клиентскую, так и серверную части, а также процесс интеграции модели нейронного переноса стиля в функционал приложения.

В начале раздела рассматриваются ключевые аспекты проектирования архитектуры веб-приложения, с акцентом на разделение задач между фронтендом и бэкендом. Разделение задач позволяет улучшить производительность приложения, сделать его более масштабируемым и поддерживаемым. Фронтенд и бэкенд веб-приложения взаимодействуют друг с другом через четко определенные интерфейсы и протоколы обмена данными, что делает их независимыми друг от друга и позволяет работать с каждой частью приложения отдельно.

Фронтенд: Фронтенд веб-приложения был разработан с использованием стандартных и современных веб-технологий: HTML, CSS и JavaScript. Использование этих технологий позволяет создать кросс-браузерное и кросс-платформенное приложение, которое работает в большинстве современных веб-браузеров и на различных устройствах.

Особое внимание уделено разработке интуитивно понятного и удобного пользовательского интерфейса. Для этого были учтены потребности пользователя: приложение должно быть простым и понятным для людей, не имеющих технических знаний. В интерфейсе предусмотрены элементы для загрузки изображений, выбора стиля и отображения результата стилизации. Основные элементы интерфейса включают кнопку для выбора изображения, кнопку для выбора стиля и область для предварительного просмотра изображений.

Структура HTML-шаблонов организована так, чтобы обеспечить удобное взаимодействие с пользователем. Важным аспектом является использование семантических HTML-элементов, что делает приложение более доступным для пользователей с ограниченными возможностями, а также улучшает индексруемость страницы поисковыми системами. Также описана структура DOM

(Document Object Model) для динамического обновления содержимого страницы без необходимости перезагрузки.

Для работы с изображениями были использованы специальные методы, такие как предварительный просмотр загруженных файлов и валидация файлов перед отправкой на сервер. Преимущество этого подхода заключается в том, что пользователь может сразу увидеть изображение, которое он собирается загрузить, и убедиться в правильности выбора. Валидация загружаемых файлов гарантирует, что на сервер будут отправляться только изображения подходящих форматов и размеров, что исключает ошибки при обработке.

Серверная часть: Серверная часть веб-приложения построена на языке Python с использованием фреймворка Flask. Flask был выбран за его минималистичность и гибкость, что позволяет быстро и эффективно разрабатывать серверную логику приложения. Он также идеально интегрируется с библиотеками для работы с нейронными сетями, такими как TensorFlow и Keras, что делает его удобным для реализации нейронного переноса стиля.

Архитектура серверного приложения включает обработку двух основных типов запросов: загрузка изображений и запуск процесса стилизации. Для обработки изображений сервер принимает изображения от пользователя через HTTP-запросы, сохраняет их с уникальными именами, чтобы избежать конфликтов и гарантировать надежное хранение данных, и затем запускает процесс нейронного переноса стиля с использованием модели VGG19.

Каждое изображение сохраняется с уникальным именем, что предотвращает возможные ошибки и потери данных при одновременной загрузке нескольких файлов пользователями. Для этого в сервере реализована система генерации уникальных имен с использованием временных меток и случайных идентификаторов, что обеспечивает надежное и безопасное хранение данных.

Интеграция модели нейронного переноса стиля: Одним из важнейших аспектов разработки является интеграция модели нейронного переноса стиля с серверной частью приложения. Модель, основанная на архитектуре VGG19, используется для извлечения признаков содержания и стиля из изображений, а затем их комбинирование для создания нового изображения, которое сочетает в себе элементы обоих компонентов.

Процесс интеграции модели в серверную часть приложения включает несколько ключевых этапов:

- **Предобработка изображений.** На сервере выполняется подготовка изображений для подачи в нейронную сеть. Это включает изменение размера изображений, нормализацию и приведение их к нужному формату с использованием функции `tf.keras.applications.vgg19.preprocess_input`.
- **Вычисление функций потерь.** Для того чтобы модель могла эффективно обучаться и минимизировать разницу между содержанием и стилем, вычисляются функции потерь по содержанию и стилю. Потеря по содержанию измеряет различие между содержимым исходного и стилизованного изображения, а потеря по стилю — насколько изображение соответствует стилю выбранного образца.
- **Оптимизация.** Процесс оптимизации заключается в минимизации комбинированной функции потерь с использованием градиентного спуска, что позволяет модели создавать изображение, сочетающее в себе характеристики как содержания, так и стиля.

Этот процесс интеграции позволяет добиться высококачественного нейронного переноса стиля, что в свою очередь значительно повышает визуальную привлекательность результирующего изображения.

Асинхронная обработка изображений: Важной частью работы серверной части является асинхронная обработка изображений, которая позволяет обеспечить высокую производительность и быстрый отклик приложения. Для этого используется механизм фоновых задач, который позволяет серверу обрабатывать несколько запросов одновременно, не блокируя основной поток.

Для реализации асинхронных задач в сервере используется фреймворк Flask с дополнением Celery, который позволяет легко интегрировать асинхронные задачи и обеспечивать их выполнение в фоне.

Взаимодействие фронтенда и бэкенда через REST API: Фронтенд и бэкенд взаимодействуют друг с другом через REST API, что обеспечивает удобный и гибкий механизм обмена данными между клиентской и серверной частями. В разделе рассматривается процесс обмена данными между фронтендом и бэкендом, который осуществляется с помощью HTTP-запросов.

Использование JavaScript и AJAX позволяет динамически обновлять интерфейс приложения без перезагрузки страницы, что значительно улучшает пользовательский опыт. При этом, с помощью асинхронных запросов, можно обновлять статус выполнения стилизации или показывать промежуточные ре-

зультаты, не блокируя взаимодействие с приложением.

Создание модели нейронного переноса стиля. В третьем разделе работы подробно рассматривается процесс создания и обучения модели нейронного переноса стиля, основанной на сверточной нейронной сети VGG19. Этот раздел охватывает все ключевые этапы разработки модели, начиная от подготовки данных и архитектуры сети до оптимизации и постобработки изображений.

В начале раздела описан процесс подготовки изображений для подачи в нейронную сеть. Этот этап включает несколько обязательных шагов, направленных на приведение изображений к формату, подходящему для обработки сверточной нейронной сетью. Важным аспектом является масштабирование изображений, которое позволяет уменьшить их размер до стандартизованных величин, подходящих для модели. Изображения изменяются по размеру, при этом сохраняется их пропорция, что предотвращает искажение исходных данных. Эта процедура выполняется с использованием стандартных библиотек Python, таких как Pillow и NumPy. Описывается алгоритм изменения размера с сохранением пропорций, а также использование функции `tf.keras.applications.vgg19.preprocess_input` для нормализации изображений, что является необходимым для корректной работы модели. Эти шаги помогают подготовить изображения для последующей обработки, увеличивая точность и качество стилизации.

Далее в разделе подробно рассматривается архитектура модели, использующей предобученную сеть VGG19. Архитектура данной сети представляет собой глубокую сверточную модель с 19 слоями, из которых 16 являются сверточными, а остальные три — полносвязными. В контексте задачи нейронного переноса стиля важно отметить, что используются только сверточные слои сети. В частности, для извлечения признаков содержания и стиля выбираются различные слои в зависимости от их глубины в сети. Признаки содержания извлекаются из более глубоких слоев сети, таких как `block5_conv2`, которые отвечают за восприятие структуры и объектов на изображении. Эти слои обеспечивают высокоуровневые признаки, которые сохраняют информацию о содержимом изображения. В отличие от этого, признаки стиля извлекаются из более поверхностных слоев, таких как `block1_conv1`, `block2_conv1`, `block3_conv1` и `block4_conv1`, которые фиксируют текстуры, цвета и узоры, характерные для художественного стиля. Активности нейронов этих слоев используются для вы-

числения потерь по содержанию и стилю, что позволяет модели эффективно совмещать оба компонента в процессе стилизации.

Ключевым элементом третьего раздела является описание функции потерь, которая используется для оптимизации процесса нейронного переноса стиля. В частности, рассматриваются три основные компоненты функции потерь: потеря по содержанию (content loss), потеря по стилю (style loss) и потеря по общей вариации (total variation loss). Потеря по содержанию измеряет различие между активациями слоев содержания для исходного изображения и генерируемого, что помогает сохранить структуру и объекты исходного изображения. Потеря по стилю вычисляется через сравнение матриц Грама, которые захватывают корреляции между активациями различных каналов признаков. Это позволяет сохранить текстурные особенности и стилистические элементы изображения. Потеря по общей вариации применяется для устранения артефактов и сглаживания изображения, улучшая его визуальное качество, устраняя шумы и резкие переходы.

После того как компоненты функции потерь были описаны, переходим к процессу оптимизации модели. Для минимизации комбинированной функции потерь используется метод градиентного спуска, который позволяет улучшить результат стилизации. Описывается использование коэффициентов α , β и γ , которые контролируют вклад каждой из потерь в итоговую функцию. Это дает возможность модели сбалансировать сохранение содержания и стиля изображения, что является ключевым для получения высококачественного результата. Особое внимание уделяется процессу итерационной оптимизации, в ходе которой изображение постепенно преобразуется в стилизованный вариант, минимизируя потери по содержанию и стилю.

После объяснения процесса оптимизации, раздел переходит к описанию этапов постобработки изображений. Этот этап необходим для восстановления изображения в исходный цветовой формат и разрешение. Также осуществляется преобразование числового массива (тензора), полученного на выходе из модели, обратно в изображение, которое можно отобразить или сохранить. Этот процесс включает денормализацию данных и изменение размера изображения до исходных пропорций. Он обеспечивает получение итогового стилизованного изображения, которое готово для отображения в интерфейсе веб-приложения.

Раздел завершается обсуждением особенностей работы с моделью в реаль-

ном времени. Рассматривается использование предобученной модели VGG19 и возможности ускорения процесса стилизации с помощью GPU. Это позволяет значительно улучшить производительность модели и сократить время обработки изображений. Важно отметить, что для работы с большими изображениями и ускорения вычислений в процессе работы можно использовать такие инструменты, как TensorFlow и CUDA, что дает возможность обрабатывать изображения значительно быстрее. Также рассматриваются возможности адаптации модели для работы с изображениями высокого разрешения, что позволяет улучшить качество результата при сохранении приемлемого времени обработки.

ЗАКЛЮЧЕНИЕ

В ходе выполнения данного дипломного проекта была реализована современная веб-платформа, предназначенная для стилизации изображений с использованием технологий искусственного интеллекта, в частности, нейронного переноса стиля на базе сверточной нейронной сети VGG19. Проект стал результатом системного и поэтапного подхода к решению сложной задачи интеграции передовых методов машинного обучения в удобный и доступный для конечного пользователя веб-сервис.

На протяжении всего процесса разработки перед нами стояли важные цели — создать приложение, которое сочетает в себе высокое качество обработки изображений и при этом остается максимально простым и интуитивно понятным для пользователей с разным уровнем технической подготовки. Важно было не только реализовать алгоритм нейронного переноса стиля, но и обеспечить стабильность, отзывчивость и адаптивность пользовательского интерфейса.

Для достижения этих задач была выполнена комплексная работа, включающая выбор и обоснование технологий, проектирование архитектуры приложения, создание удобного и современного интерфейса, а также оптимизацию серверной части, обеспечивающей быструю и надежную обработку пользовательских данных. Особое внимание уделялось технической стороне — использованию эффективных алгоритмов и инструментов, позволяющих добиться высокого качества стилизации при приемлемом времени обработки.

В процессе разработки был реализован адаптивный дизайн, который гарантирует корректное отображение и работу приложения на самых разных устройствах — от полноразмерных мониторов до мобильных телефонов и планшетов. Это существенно расширяет аудиторию потенциальных пользователей и повышает доступность сервиса.

Кроме того, была проделана работа по обеспечению безопасности и надежности приложения. Внедрение современных методов логирования, обработки ошибок и асинхронной обработки задач позволило создать устойчивый к нагрузкам и ошибкам сервис, готовый к дальнейшему развитию и масштабированию.

Результатом проекта стало веб-приложение, которое полностью отвечает заявленным требованиям и задачам. Оно предоставляет пользователям возможность быстро и просто преобразовывать изображения, используя сложные алго-

ритмы нейронного переноса стиля, без необходимости разбираться в тонкостях машинного обучения и программирования.

Стоит отметить, что данный проект открывает широкие перспективы для дальнейшего развития и совершенствования. В будущем планируется расширение функционала, включающее поддержку более разнообразных моделей стилизации, улучшение скорости обработки за счет использования аппаратного ускорения и облачных технологий, а также интеграцию с социальными платформами для упрощения обмена результатами.

Таким образом, выполненная работа не только подтверждает применимость современных методов искусственного интеллекта в реальных пользовательских приложениях, но и демонстрирует, как можно эффективно сочетать научные достижения с практическими задачами создания удобных цифровых продуктов. Полученное приложение может стать полезным инструментом для широкого круга пользователей — от любителей фотографии до профессиональных дизайнеров, открывая новые возможности для творчества и самовыражения в цифровом пространстве.

В завершение, можно с уверенностью сказать, что поставленные в начале работы цели достигнуты, а реализованный продукт обладает высоким потенциалом для дальнейшего развития и применения в различных сферах деятельности.