

РАЗРАБОТКА ПРОГРАММНОГО СРЕДСТВА ДЛЯ РЕАЛИЗАЦИИ ФУНКЦИИ ТРАНСФОРМАЦИИ ИЗОБРАЖЕНИЙ ПО АВТОРСКОМУ СТИЛЮ НА ОСНОВЕ АЛГОРИТМОВ МАШИННОГО ОБУЧЕНИЯ

DEVELOPMENT OF A SOFTWARE TOOL
FOR REALIZING THE FUNCTION
OF IMAGE TRANSFORMATION
BY AUTHOR'S STYLE BASED ON MACHINE
LEARNING ALGORITHMS

**A. Rusakov
V. Solovyova
I. Rakhimov
I. Vorobyov
V. Filatov**

Summary. This article presents a study on the development of a tool for realizing the function of image transformation by author's style on the basis of machine learning algorithms. The research materials contain a detailed description of the choice of the architecture of the developed system, its implementation features, as well as the description of the main modules and functions of the developed software. The main tasks are formulated, methods and algorithms to be considered in the development are identified.

Keywords: image transformation, machine learning.

Русаков Алексей Михайлович

старший преподаватель, МИРЭА — Российский
технологический университет
rusal@bk.ru

Соловьева Вероника Антоновна

МИРЭА — Российский технологический университет
solovieva.veronika00@mail.ru

Рахилев Игорь Русланович

МИРЭА — Российский технологический университет
rakhilov@bk.ru

Воробьев Иван Алексеевич

МИРЭА — Российский технологический университет
vorobev_i_a@mail.ru

Филатов Вячеслав Валерьевич

кандидат технических наук, доцент,
МИРЭА — Российский технологический университет
filv@mail.ru

Аннотация. В данной статье представлено исследование по разработке инструмента для реализации функции трансформации изображений по авторскому стилю на основе алгоритмов машинного обучения. Материалы исследования содержат подробное описание выбора архитектуры разрабатываемой системы, ее особенности реализации, а также описание основных модулей и функций разрабатываемого программного обеспечения. Сформулированы основные задачи, выявлены методы и алгоритмы, которые необходимо учесть при разработке.

Ключевые слова: трансформация изображений, машинное обучение.

На сегодняшний день обработка медиа данных, в зависимости от того, статические это изображения, видео или даже онлайн видео — трансляции в реальном времени, является очень актуальной темой. Объемы медиа растут, соответственно, необходимость в их обработке также увеличивается. Многие из нас используют средства обработки изображений и видео в своей повседневной жизни, ведя блог в Instagram, Twitter, Facebook или YouTube. Такие параметры как скорость и автоматизация процесса обработки медиа делают подобные сервисы удобным и легким в использовании, но в то же время накладывают определенные ограничения.

Однако за помощью в подобных ситуациях можно обратиться к специализированному профессиональному программному обеспечению обработки медиа данных, например, такому как Adobe Photoshop, Sony Vegas Pro

и др. Но использование подобных программных средств требует определенных навыков работы, поскольку имеет под собой ручную обработку и редактирование данных, что повышает для конечных пользователей порог вхождения и увеличивает трудозатраты в несколько раз.

Средством автоматизации определенных рутинных действий пользователя по обработке изображений и видео могут выступать алгоритмы машинного обучения, а именно — нейронные сети. С помощью нейронных сетей можно автоматизировать такие процессы обработки изображений как комплексное улучшение качества изображения, цветокоррекция, наложение стилей и их трансформацию, наложение масок и многое другое, избежав при этом без необходимости ручного редактирования и коррекции медиа данных.

Таким образом, целью данной работы является создание удобного в использовании для пользователя про-

граммного средства автоматизации художественной обработки изображений методами машинного обучения.

Выбор архитектуры разрабатываемой системы

Поскольку разрабатываемая система прежде всего является программным средством трансформации стиля, при его создании может быть использована абсолютно любая архитектура, ведь в основе лежит алгоритм и средства его обработки, которые являются одинаковыми для любой платформы — Web, Desktop, Mobile или Embedded. Поэтому, разработанное ПО должно быть цельным ядром машинного обучения, которое имеет программный интерфейс (API), созданный таким образом, что будет обеспечено поэтапное выполнение следующей логики работы: запуск с входным изображением или видео, обработка, завершение с исходным трансформируемым изображением или видео.

Трансформатор стиля можно разместить на любой платформе, добавив к нему дополнительное программное обеспечение управления — сервер, плагин или приложение-интерфейс.

Клиент-серверная архитектура подойдет для организации на Web-платформе. Для этого требуется создать среду в виде контейнера Docker с необходимой операционной системой и разместить на нем трансформатор стиля. В свою очередь сервер будет осуществлять управление трансформатором через его программный интерфейс и веб-страницу для доступа и управления пользователем. Соответственно, хостинг должен поддерживать внешние статические адреса. Возможно также совместить трансформатор стиля и сервер в одном процессе, однако это уменьшит общую гибкость системы.

Следующим вариантом платформы является Desktop и Mobile. С архитектурной точки зрения они абсолютно идентичны. Обе имеют под собой высокоуровневую разработку под определенную операционную систему. Так же, как и в Web-платформе, трансформатор стиля будет запускаться как дочерний процесс основного приложения, которой и будет включать в себя графический интерфейс пользователя и всю необходимую бизнес-логику. Такая архитектура называется межпроцессным взаимодействием (рис. 1) [12].

Заключительный вид платформы — это Embedded. Она подразумевает под собой разработку системы на низком аппаратном уровне. В этом также все сводится к межпроцессорному взаимодействию за исключением того, что ресурсы крайне ограничены в совокупности с инструментарием разработки, однако производительность такой системы является высокой.

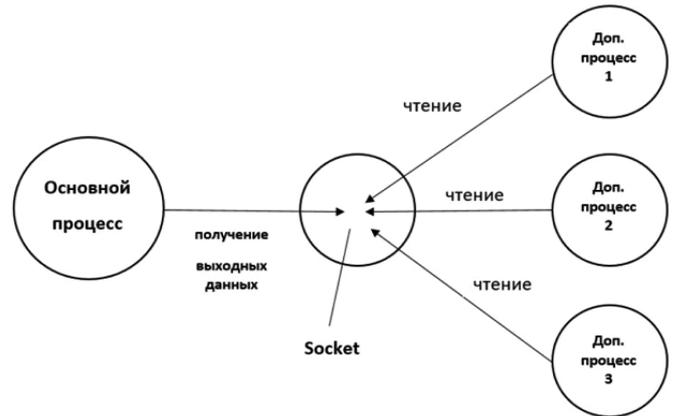


Рис. 1. Общая схема межпроцессорного взаимодействия

Грамотным решением будет использование аппаратного ускорения для трансформатора стиля, в независимости от того, будет это мощный графический процессор или любой специализированный AI-ускоритель.

Исходя из требований кроссплатформенности разрабатываемого трансформатора стиля и простоты реализации готового программного средства, демонстрируя его работу, для разработки была выбрана Desktop платформа. Готовое программное обеспечение должно состоять из кроссплатформенного трансформатора стиля, который будет иметь собственный программный интерфейс и осуществлять запуск благодаря дочернему процессу GUI-приложения. В качестве целевой операционной системой была выбрана Windows.

Особенности реализации системы

К основным особенностям реализации данной системы относятся:

- автономность трансформатора стиля (подразумевает под собой простой и удобный программный интерфейс, в то же время управляющее программное обеспечение может быть любым);
- применение TensorFlow в качестве фреймворка машинного обучения;
- использование аппаратного ускорения на GPU благодаря NVIDIA CUDA и cuDNN;
- использование Qt Framework для создания удобного и стилизованного GUI;
- использование трансформатора стиля в качестве дочернего процесса конечной программной реализации для пользователя;
- возможность трансформации как изображений (поддерживаются форматы PNG, JPG, BMP), так и видео (поддерживается формат MP4) и возможность одновременного просмотра входных и выходных данных для наглядности;
- в качестве входящих и исходящих данных выступает изображение или видео.

Разработка основных алгоритмов программного обеспечения

В сумме в данном модуле представлено 3 основных алгоритма — трансформация изображения, трансформация видео, а также тренировка сети. Подробная схема алгоритмов обучения СНС и трансформации приведена в первой главе с описанием самой модели сети. Ниже изложены лишь последовательность вызова процедур модулей для каждого из алгоритмов на высоком уровне абстракции (рис. 2) (Выполнено в бесплатном онлайн редакторе блок-схем DrawIO). Также стоит обратить внимание, что реализация алгоритмов в коде выглядит довольно простой и небольшой по объему только благодаря высокоуровневому API TensorFlow Estimators.

При этом сама программа не имеет строгого алгоритма выполнения. Есть только определенные потоки вы-

полнения, которые может выбрать пользователь для достижения своих целей, демонстрируемых диаграммой прецедентов, приведенной ниже (рис. 3) (Выполнено в бесплатном онлайн UML редакторе Paradigm).

Разработка структуры программного обеспечения

Модуль *ml_core*

Данный модуль представляет собой непосредственно программное средство, реализующее трансформацию стилей различных изображений и видео. Это набор скриптов на Python, в которых используются вспомогательные модули, такие как: модуль тренировки сети и модули трансформации изображений и видео. Ниже представлена общая структура модуля *ml_core*.

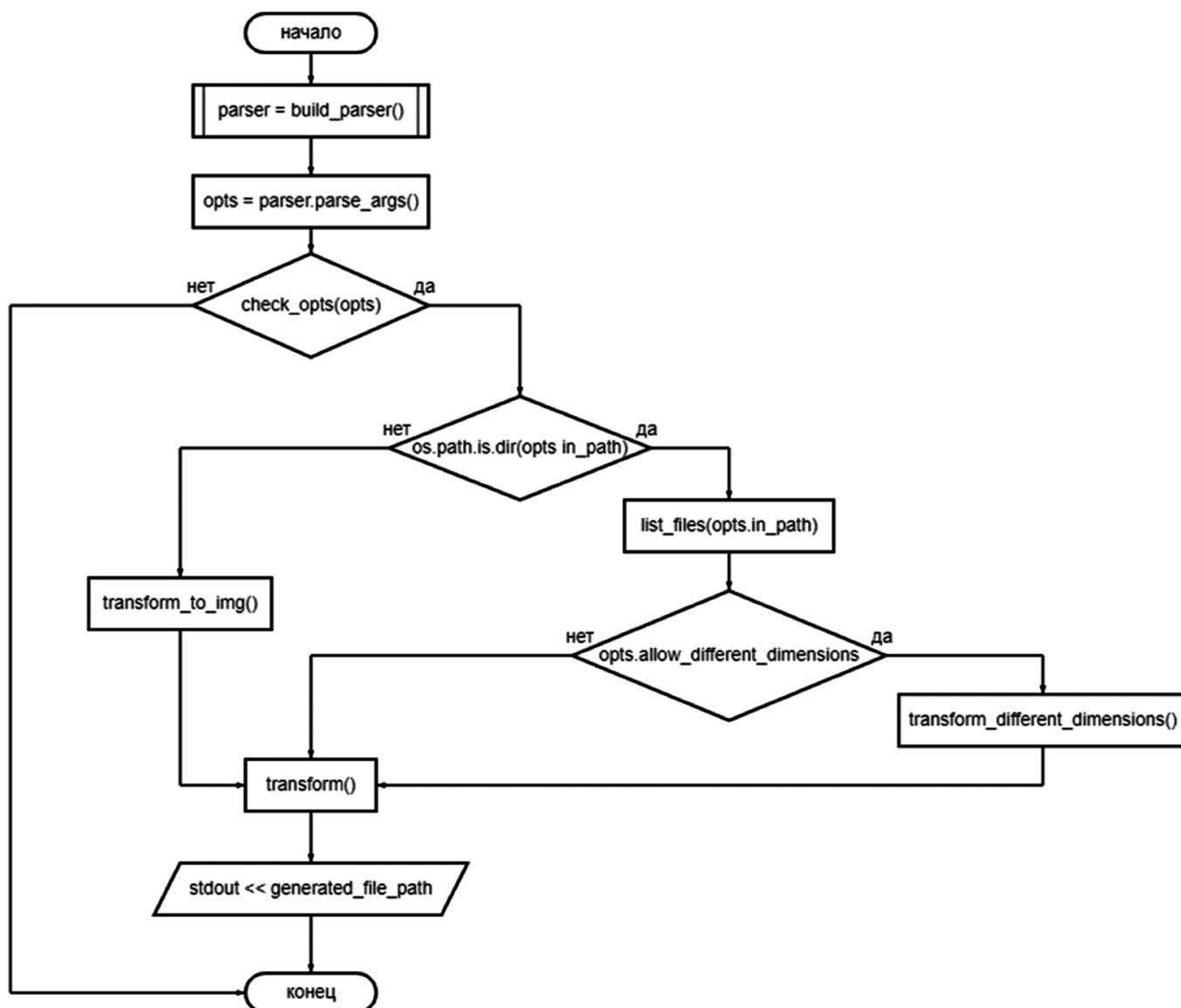


Рис. 2. Алгоритм трансформации изображения

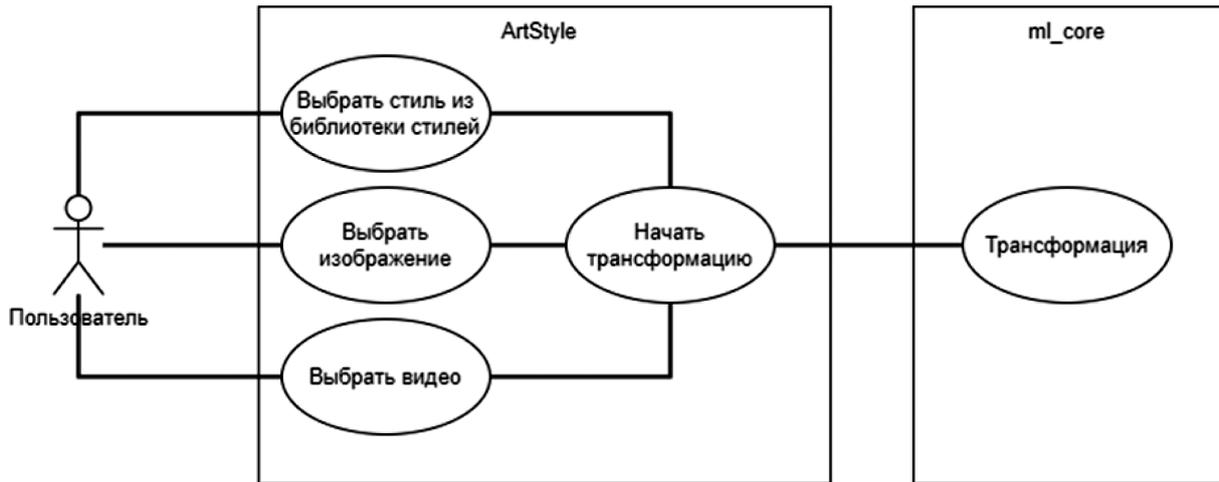


Рис. 3. Диаграмма прецедентов ArtStyle

Исполнительные модули:

- train.py — запускает тренировку новой сети, настроенной под определенный стиль;
- transform_img.py — запускает трансформацию изображения;
- transform_video.py — запускает трансформацию видео;

Вспомогательные модули:

- optimize.py — представляет функцию-оптимизатор для тренировки сети (тренировочная функция);

- transform_cnn.py — представляет СНС;
- utils.py — набор функций-утилит;
- vgg.py — представляет VGG19 сеть.

При разработке данного компонента была использована только процедурная парадигма программирования с классической модульной архитектурой, поскольку алгоритм является линейным и структура модуля достаточно проста. Присутствует выделение главных модулей, которые являются исполнительными и вспомогательных подмодулей (рис. 4) (Выполнено в бесплатном онлайн UML редакторе Paradigm).

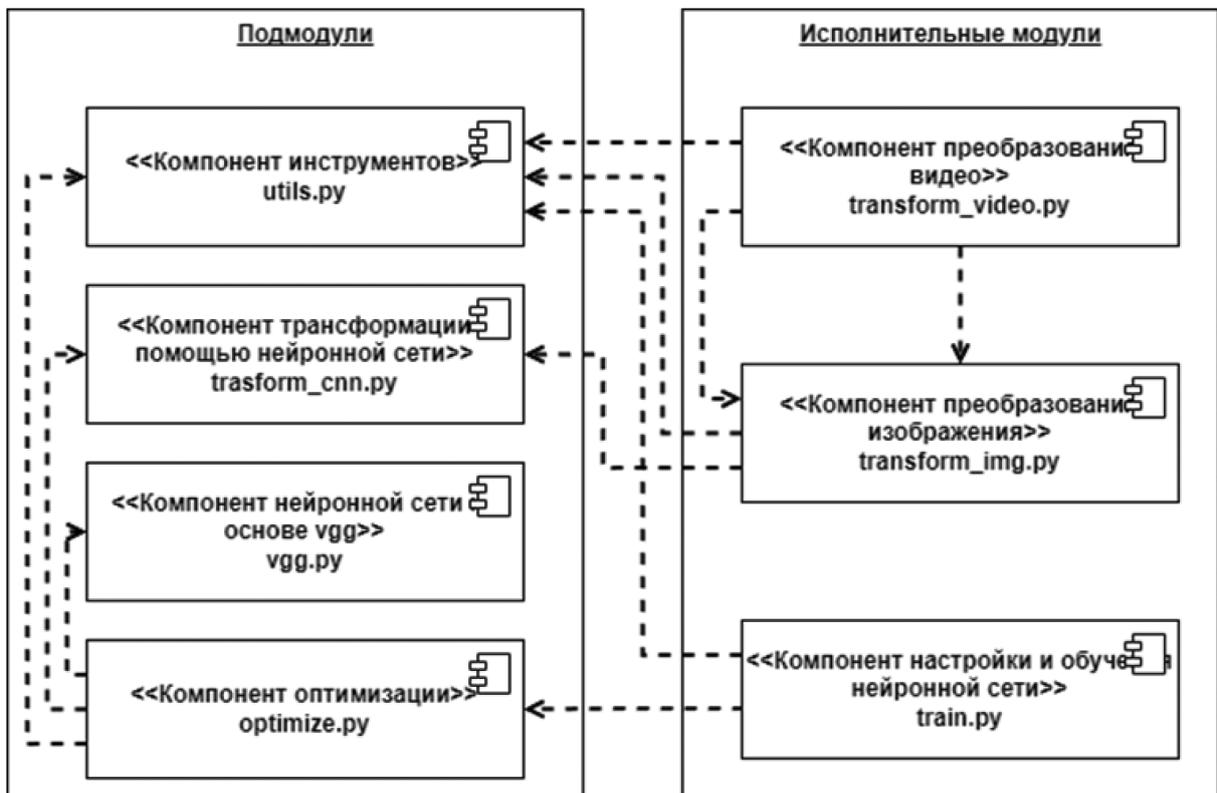


Рис. 4. Диаграмма компонентов ml_core

Модуль qt_gui

Модуль представляет собой GUI-часть для ml_core. Это десктопное приложение, написанное на C++ с использованием Qt Framework, которое демонстрирует базовый пример использования трансформатора стиля. Это один из возможных вариантов реализации пользовательского интерфейса для трансформации стилей изображения и видео. Ниже описана структура программы.

Программа состоит из следующих классов:

- `QStyleTransformator` (`QStyleTransformator.h` и `QStyleTransformator.cpp`) — класс, представляющий все приложение как логическую структуру. Его инициализация и запуск означает запуск всей программы. Он создает замкнутую логическую среду выполнения программы. Инициализируется в `main()` функции, существует там и там же и уничтожается;
- `QMainWnd` (`QMainWnd.h` и `QMainWnd.cpp`) — класс, представляющий собой главное окно программы. Инкапсулирует GUI-часть программы и создает среду для размещения и функционирования компонентов графического интерфейса. `QMainWnd.ui` — UIC-разметка главного окна;
- `QImageViewWdgt` (`QImageViewWdgt.h` и `QImageViewWdgt.cpp`) — потомок класса `QWidget`. Представляет собой собственный разработанный виджет, предназначенный для просмотра изображений. Размещается на главном окне программы. Инкапсулирует в себе весь функционал, связанный с графическим представлением изображения на форме;
- `QVideoPlayerWdgt` (`QVideoPlayerWdgt.h` и `QVideoPalyerWdgt.cpp`) — потомок класса `QWidget`. Представляет собой собственный разработанный виджет, предназначенный для просмотра видео. Размещается на главном окне программы. Инкапсулирует в себе весь функционал, связанный с графическим представлением видео на форме;
- `QTransformApiHandler` (`QTransformApiHandler.h` и `QTransformApiHandler.cpp`) — класс, который имплементирует API трансформатора стиля `ml_core`. Содержит в себе и контролирует дочерний процесс `ml_core`;

Вспомогательные структуры:

- `SConfig` (`SConfig.h` и `SConfig.cpp`) — представляет конфигурацию программы и функционал загрузки. Формат конфигурационных данных — XML;
- `SContext` (`SContext.h`) — представляет глобальный контекст программы. Содержит в себе конфигурацию и определения типов трансформации (в виде `enum`).

Описание основных функций программного обеспечения

Разрабатываемая система состоит из двух блоков:

1. Непосредственно само программное средство трансформации стиля, написанное на языке программирования (ЯП) Python (далее `ml_core` модуль).
2. GUI-часть, написанная на ЯП C++ с использованием Qt Framework (далее `qt_gui` модуль).

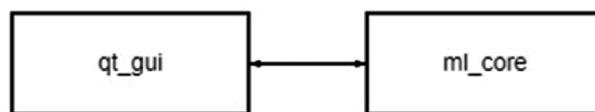


Рис. 5. Общая схема разрабатываемой системы

Каждый из представленных компонентов представляет собой отдельный процесс операционной системы, который в свою очередь взаимодействует с другими аналогичными процессами. Главным процессом является `qt_gui`, дочерним — `ml_core`.

Компонент `qt_gui` представляет собой простое GUI-приложение, реализующее весь основной функционал `ml_core` путем имплементации API последнего. В его состав входят следующие блоки:

- библиотека стилей;
- панель просмотра и трансформации изображений;
- панель просмотра и трансформации видео.

`ml_core` как отдельный компонент системы представляет собой консольное приложение, которое принимает на входе стиль и изображение или видео с дополнительными параметрами, а на выходе предоставляет трансформированное изображения или видео. Данный модуль состоит из следующих компонентов:

- `transform_img.py` — скрипт для запуска трансформации изображения;
- `transform_video.py` — скрипт для запуска трансформации видео;
- `train.py` — скрипт для запуска тренировки нового стиля.

Функционал `qt_gui` модуля сводится к представлению удобного графического интерфейса пользователя. Он лаконичный с эргономичным дизайном, представлен в удобной форме, а также позволяет выполнять основную функцию модуля `ml_core` — наглядную трансформацию изображений и видео.

Пример работы программного обеспечения

На рис. 6 представлено главное окно программы с открытой галереей стилей (рис. 6).

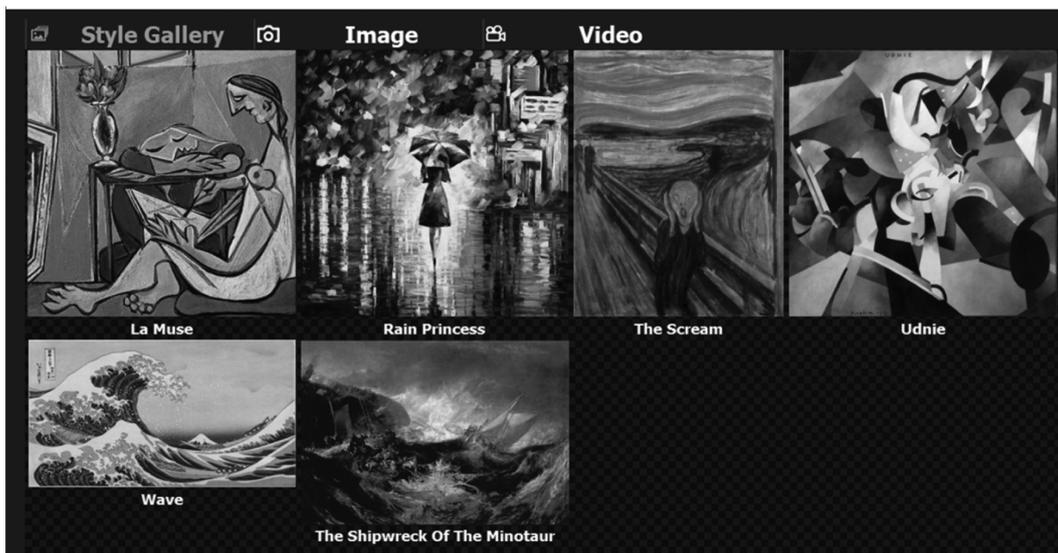


Рис. 6. Главное окно программы и панель галереи стилей

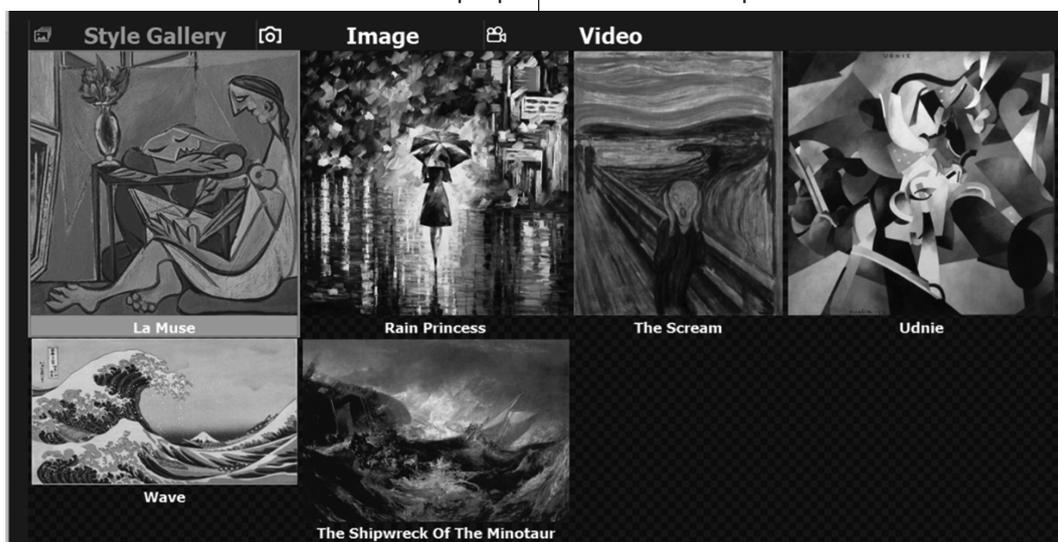


Рис. 7. Выбранный стиль «La Muse» из галереи стилей

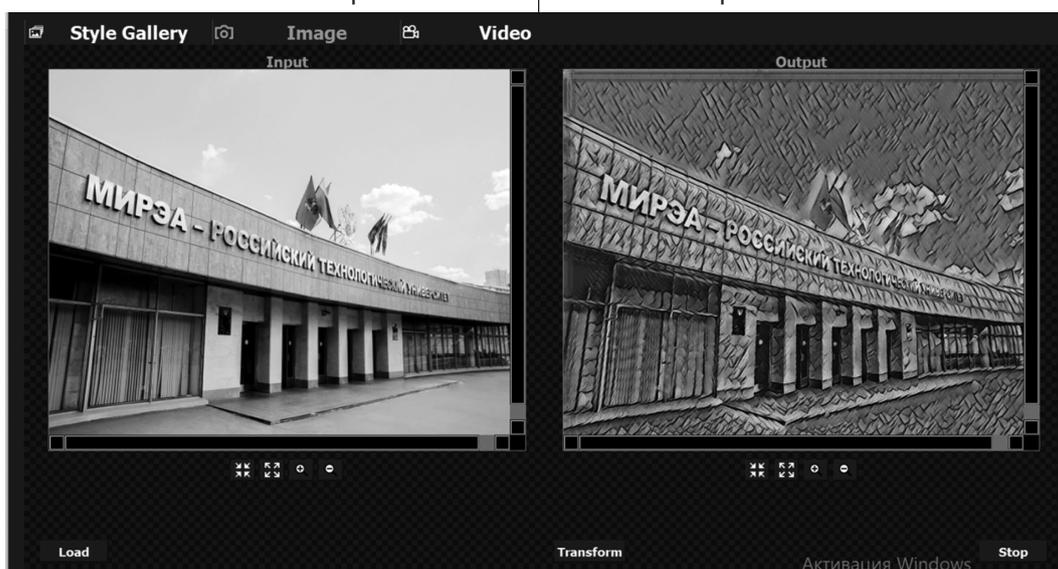


Рис. 8. Трансформация изображения завершена

Для начала трансформации стиля, необходимо сначала выбрать стиль из галереи стилей (рис. 7).

После данного этапа, для трансформации изображений необходимо перейти на соответствующую панель, которая содержит два обозревателя изображений, для входящего и исходящего файла (рис. 8).

Далее требуется загрузить входное изображение, нажав кнопку «Load». Появится диалоговое окно выбора файла, в котором необходимо выбрать файл изображения, после чего обозреватель входного изображения будет заполнен загруженным изображением. Масштаб изображения можно регулировать кнопками управления обозревателем.

Для начала трансформации изображения, необходимо нажать кнопку «Transform» на панели изображений. Внизу появится индикатор выполнения в виде зеленой ленты прогресса. Для прерывания процесса трансформации необходимо нажать кнопку «Stop».

После завершения трансформации автоматически заполнится обозреватель исходного изображения и исчезнет индикатор прогресса. Трансформация изображения считается завершенной.

Для трансформации видео необходимо перейти на соответствующую панель, которая содержит два видео плеера, для входящего и исходящего соответственно. Сразу после запуска ArtStyle.exe видео плееры пустые.

Далее нужно загрузить входящее видео нажав кнопку «Load». Появится диалоговое окно выбора файла, в кото-

ром необходимо выбрать файл видео, после чего видео плеер входящего видео будет заполнено загруженным видеороликом. Плеером можно управлять с помощью его панели управления.

Для начала трансформации видео, необходимо нажать кнопку «Transform» на панели видео. Внизу появится индикатор выполнения в виде зеленой ленты прогресса. Для прерывания процесса трансформации необходимо нажать кнопку «Stop».

После завершения трансформации автоматически заполнится выходной видео плеер и исчезнет индикатор прогресса (рис. 9). Трансформация видео считается завершенной.

Выводы

В результате проведенной работы, с помощью среды программирования MS Visual Studio 2022, Qt Designer, Visual Studio Code и языков программирования C ++, Python с использованием фреймворков Qt и TensorFlow, было создано программное обеспечение для трансформации стиля изображений и видео в виде двух модулей — модуль трансформации стиля и модуль клиента, управляющий трансформацией.

Разработанная система позволяет:

- просматривать имеющиеся для наложения стили;
- трансформировать стиль изображений;
- трансформировать стиль видео.

Разработанная система может улучшаться в будущем путем добавления новых стилей, оптимизации трансформирующего алгоритма, а также переноса модуля



Рис. 9. Трансформация видео завершена

трансформации на язык C ++, что должно значительно повысить скорость работы системы.

Также возможна более общая реализация системы путем разработки сервера управления модулем транс-

формации стилей и его графической интерфейсной части в виде веб-страницы. Другими словами — реализация в виде веб-сервиса.

ЛИТЕРАТУРА

1. Гонсалес Р., Вудс Р. Цифровая обработка изображений. — 2012 [Электронный ресурс]. URL: <https://studizba.com/files/show/pdf/84807-1-gonsales-r-vuds-r-cifrovaya-obrabotka.html>.
2. Нестереня И.Г., Махнач Н.С. Применение нейронных сетей для решения творческих задач. — 2017 [Электронный ресурс]. URL: <https://libdoc.bsuir.by/handle/123456789/27623>.
3. Жерон О. Прикладное машинное обучение с помощью Scikit-Learn и TensorFlow ШШ. — 2018 [Электронный ресурс]. URL: https://fileskachat.com/view/69498_fb6bc614e748510c39dd2d649a9443f3.html
4. Зулунов Р.М., Солиев Б.Н. ИСПОЛЬЗОВАНИЕ PYTHON ДЛЯ ИСКУССТВЕННОГО ИНТЕЛЛЕКТА И МАШИННОГО ОБУЧЕНИЯ //AI-Farg'oniy avlodlari. — 2023. — Т. 1. — №. 3. — С. 18–24 [Электронный ресурс]. URL: <https://journals.indexcopernicus.com/search/article?articleId=3712460>.
5. Нарбут А.П., Либеров Р.В. Методы стилизации изображений на основе свёрточной и глубокой нейронной сети //Электронный журнал: наука, техника и образование. — 2017. — №. 1. — С. 27–30 [Электронный ресурс]. URL: <https://nto-journal.ru/catalog/informacionnye-technologii/248/>.
6. Ульянов Д.И., Савельев Д.А. Исследование применения циклических генеративно-состязательных нейронных сетей для стилизации изображений //Информационные технологии и нанотехнологии (ИТНТ-2020). Сборник трудов по материалам VI Международной конференции и молодежной школы (г. Самара, 26–29 мая). — 2020. — №. 4. — С. 198–205 [Электронный ресурс]. URL: <http://repo.ssau.ru/handle/Informacionnye-tehnologii-i-nanotehnologii/Primenenie-svertochnyh-neironnyh-setei-dlya-ustanovleniya-vozhrasta-cheloveka-po-izobrazheniu-84903>.
7. Гольдштейн М.А. Использование сверточных нейронных сетей в задачах стилизации и синтеза текстур //Аллея науки. — 2017. — Т. 2. — №. 9. — С. 849–855 [Электронный ресурс]. URL: <http://lib.tssonline.ru/articles2/fix-corp/postroenie-arhitektury-neyronnoy-seti-dlya-vyyavleniya-vida-raspredeleniya-sluchaynyh-velichin>.
8. Бэрри П. Изучаем программирование на Python. — Litres, 2019 [Электронный ресурс]. URL: <https://clck.ru/3AArou>.

© Русаков Алексей Михайлович (rusal@bk.ru); Соловьева Вероника Антоновна (solovieva.veronika00@mail.ru);
Рахилов Игорь Русланович (rakhilov@bk.ru); Воробьев Иван Алексеевич (vorobev_i_a@mail.ru); Филатов Вячеслав Валерьевич (filv@mail.ru)
Журнал «Современная наука: актуальные проблемы теории и практики»