

Пояснительная записка

к итоговому проекту на тему:

# **“Стилизация изображений с помощью нейронных сетей, как метод современного изобразительного искусства”**

# 

Автор: Горшков Алексей

Группа: DSU-55

# ОГЛАВЛЕНИЕ

1. Постановка задачи …………………………..…………………………………………….3
2. Анализ и предобработка ……………………………………….………………………….4
3. Методика реализации………………………………………………………………..…….7
4. Telegram бот…………..……………………………………………………………..…….10
5. Итоги обучения……………………………………………………………………………11
6. Выводы …………………………………………………………………………………….18
7. Список литературы……..………………………………………………………………...19

Постановка задачи

Стилизация встречается во многих сферах, связанных с творчеством, изобразительным искусством, музыкой, фотографией, дизайном. Она везде, где есть стиль.

Стилизация помогает передать творческий взгляд через художественный образ и создать уникальный неповторимый и узнаваемый стиль. Придать художественную выразительность или сделать отсылку к значимому событию, известному образу или продукту. Постер в стиле барокко, мозаика под модерн, историческая отсылка к древней культуре.

Понятие «Стилизация», объясняется как декоративное обобщение форм при помощи ряда определенных приемов:

- Упрощение. Отбрасываем лишнее, убираем визуальный шум и делаем мотив наиболее понятным. Выделение формы объекта и его конструктивные особенности.

- Обобщение. Помогает упростить форму и восприятие. Обобщить можно подобные по форме или цвету предметы, малые формы в большие, целую форму до силуэта.

- Деформация. Изменение формы объекта. Объект можно увеличить, уменьшить, разбить на составляющие, изменить их расположение.

- Абстрагирование. Выделение основ формообразования с отвлечением от случайных качеств.

- Геометризация. Стилизация объекта на основе геометрических форм и фигур: квадратов, кругов, кубов, цилиндров.

Современные технологии нейронных сетей и алгоритмов глубокого машинного обучения открывают новые возможности для стилизации изображений. Перенос стиля (style transfer) - одно из наиболее креативных особенностей сверточных нейронных сетей. Взяв контент с одного изображения и стиль от второго, нейронная сеть объединяет их в одно новое уникальное художественное произведение. Передача нейронного стиля основана на концепции глубоких нейронных сетей, которые способны изучать и представлять сложные закономерности в данных. Используя эти закономерности, алгоритм нейронной сети может извлекать особенности стиля и содержания изображения и объединять их с другим изображением.

На сегодня представлено множество программ, сервисов и приложений с использованием алгоритмов нейронных сетей, который умеют преобразовывать и стилизовать изображения. Но для одних необходима регистрация, доступ к компьютеру или установка платных сторонних программ. А так, же небольшой заранее предустановленный набор стилей.

Мобильные гаджеты, стали неотъемлемой частью нашей жизни. В них хранится многое о нас и то, что с нами связано. Это уже больше, чем просто мобильный телефон для звонков. И почти все фотографии делаются с начала на телефон, а потом показываются друзьям, пересылаются в чаты и возможно даже печатаются для фотоальбомов.

Именно поэтому был сделан выбор на создании приложения для стилизации изображений и фотографий, с быстрым и удобным доступом через мобильное устройство. Сделал фотографию - и через несколько минут получаешь ее в новом стилизованном представлении.

Анализ и предобработка данных

Для реализации этой задачи не будет применен «классический» подход для анализа данных - процесс проверки, очистки, преобразования, моделирования данных, выявления закономерностей и выбросов. Так как и не будет датасета - заранее подготовленного и размеченного набора данных.

В качестве исходных данных, из открытых источников, были собраны изображения, объединенные одной стилистической темой. Выбрано семь тематических групп: абстракционизм, граффити, гризайль, импрессионизм, кубизм, постимпрессионизм, поп-арт. Каждая группа состоит из трех характерных изображений стилевой направленности.

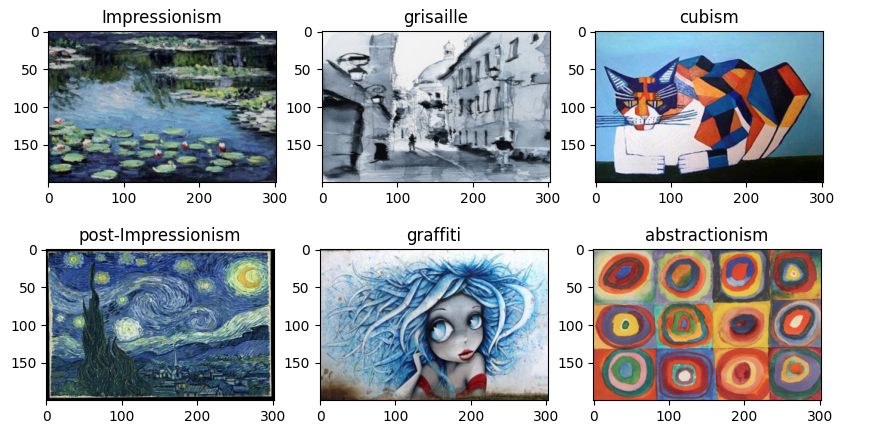


Рис.1. Примеры изображений стиля.

Для передачи изображений стиля в модель нейронной сети, была произведена предобработка входных данных. Все картинки были переведены в один формат и преобразованы к одному размеру. Затем изображения переведены в тензоры.

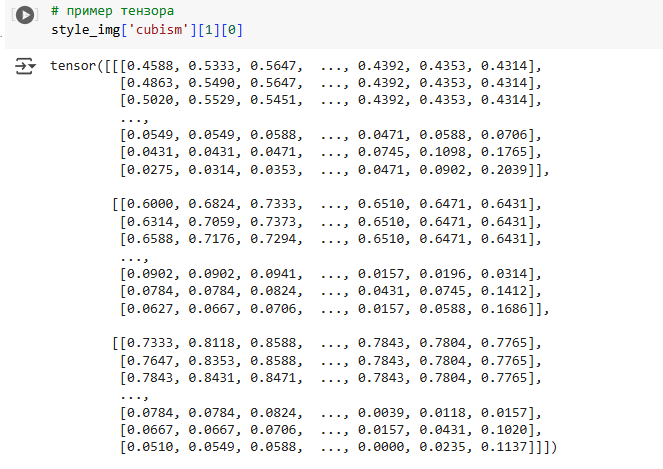


Рис.2. Примеры тензора изображения стиля.

Все полученные изображения-тензоры упакованы в словарь для удобства и быстроты выбора нужного стиля и работы модели нейронной сети.

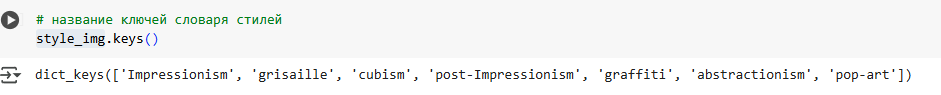


Рис.3. Словарь изображений-тензоров стиля.

Методика реализации

VGG – это свёрточная нейронная сеть с глубиной 19 слоев. Она была построена и обучена К. Симоняном и А. Зиссерманом в Оксфордском университете в 2014 году. Сеть VGG-19 обучена с использованием более одного миллиона изображений из базы данных ImageNet. Она обучалась на цветных изображениях размером 224x224 пикселей. Эта предварительно обученная сеть может классифицировать до тысячи объектов.

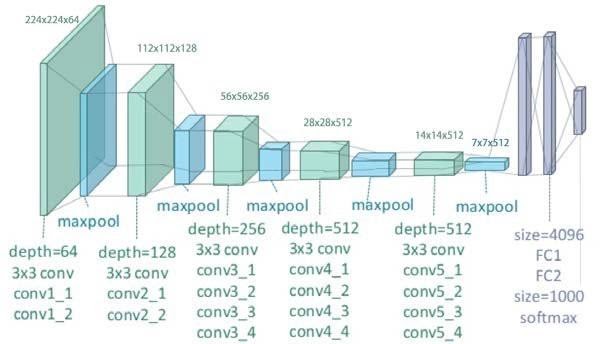


Рис.4. Иллюстрация сети VGG-19.

Задача по передачи стиля заключается в определении двух функций расстояния. Одна из них описывает, насколько друг от друга отличаются содержания двух изображений (L - content). Вторая функция описывает разницу между двумя стилями изображений (L - style). В модель передается три изображения (стиль, контент и входное изображение). Нейронная сеть пытается преобразовать входное изображение так, чтобы минимизировать его расстояние L - content с изображением контента и расстояние L - style с изображением стиля.

Важным аспектом для стилизации изображения с помощью нейронной сети является получение некоторого представления картинки стиля и картинки контента. Эти представления скрыты в промежуточных слоях модели. Промежуточные слои модели представляют собой карты признаков, которые по мере углубления становятся более упорядоченными. Сверточные нейронные сети могут хорошо обобщать: они способны заметить постоянство и определить особенности, характерные для какого-либо класса, не обращая внимания на фон и шум. Между подачей изображения на вход и выводом результата классификации этого изображения, модель находит признаки во входных данных. И обращаясь к некоторой промежуточной точке, промежуточному слою, можно получить представление стиля и содержания картинки.



Рис.6. Архитектура VGG-19.

Процесс переноса стиля с одного изображения другому заключается в определении расстояния между двумя промежуточными представлениями этих изображений.

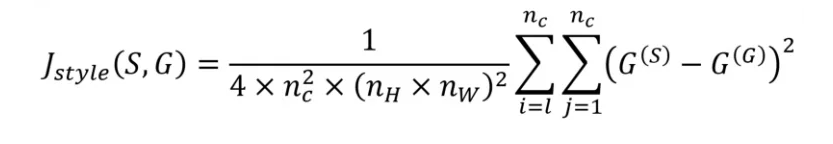
Определение функции потерь для контента:

Полученное сгенерированное изображение моделью должно соответствовать содержимому входного изображения. Другими словами, генерируемое изображение также должно иметь те же объекты и предметы (окружение, здания, облака, и т.д.). Следовательно, функция потерь для контента может быть определена как тензор квадратов разностей между признаками контентного изображения и формируемого.



2. Определение функции потерь для стиля:

Определение похожести стиля между исходным изображением и сгенерированным моделью заключается в сравнении расстояний двух матриц (матрица Грама). Матрица Грама представляет собой скалярное произведение точек набора векторов. Это отражает сходство между каждым вектором, если два вектора похожи друг на друга, то скалярное произведение их точек будет большим. А, следовательно, матрица Грама будет иметь большие значения. Функция потерь для стиля, определена как квадрат разности между двумя матрицами Грама, стиля сгенерированного изображения и исходного изображения стиля. Другими словами, мы хотим минимизировать расстояние между двумя матрицами Грама.



1. Функция оптимизации.

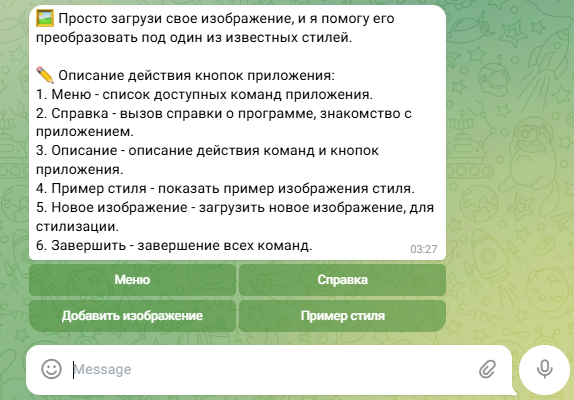
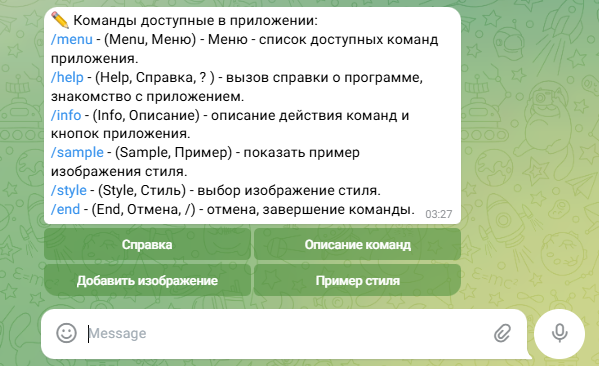
Для подсчета потери градиента используем оптимизатор Adam. При каждом проходе модели будем подсчитывать и обновлять градиенты слоя контента и слоя стиля.

1. Дополнительные контролируемые параметры.

В качестве дополнительных параметров отметим время работы модели. Время для преобразования исходного изображения должно быть не сильно большим. Создаваемое приложение, должно работать в режиме реального времени, и долгое ожидание может вызвать негативную реакцию у пользователя. Так же будем визуально отслеживать изменение и преобразование исходного изображения.

Telegram бот

Приложение для взаимодействия с пользователем реализовано на базе Telegram – бота. Для взаимодействия с пользователя с ботом настроена удобная навигация, с подсказами и описанием. Выведены функциональные клавиши и команды.



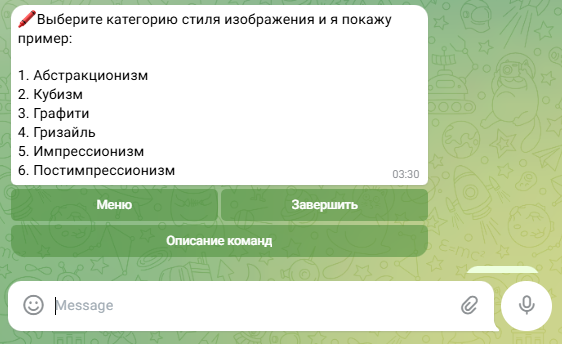
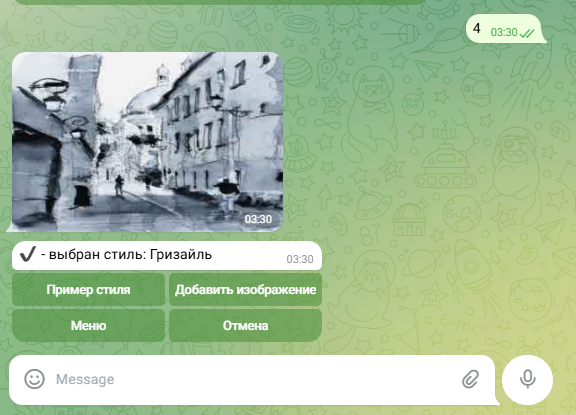
 

Рис.7. Интерфейс Telegram-бота

Пользователь загружает свое изображение и выбирает одну из категорий стиля. Далее бот выдаст информационное сообщение и начнет работу по стилизации картинки. Так же можно посмотреть примеры стилевых изображений, которые загружены в Telegram-бота.

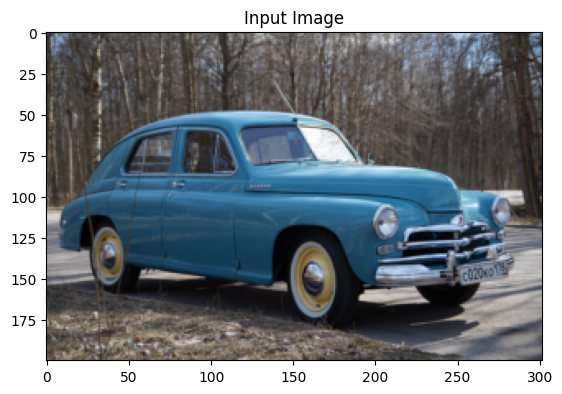
Итоги обучения

Реализация модели сверточной нейронной сети и эксперименты с по переносу стиля изображения проводились в Google Colab. Colab позволяет получать построчно результаты работы модели. Тестирование модели CNN в Google Colab проводились без подключения графического

процессора. Запуск приложения планируется на локальном компьютере, где нет технической возможности подключить графический процессор GPU.

Результаты экспериментов:

1. Изображение контент:



1. Изображение Style:



1. Стилизованное изображение:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Шаг итерации (N step)** | **Значение ошибки стиля (Score style)** | **Значение ошибки контента**  **(Score content)** | **Время работы**  **(сек.)** | **Изображение** |
| 50 | 152 339 | 78 | 63 |  |
| 100 | 80 334 | 122 | 125 |  |
| 150 | 52 637 | 141 | 180 |  |
| 200 | 38 237 | 152 | 238 |  |
| 250 | 29 309 | 161 | 304 |  |
| 300 | 23 277 | 167 | 358 |  |
| 350 | 19 012 | 172 | 417 |  |
| 400 | 15 875 | 177 | 477 |  |
| 450 | 13 492 | 181 | 537 |  |
| 500 | 11 625 | 184 | 595 |  |
| 550 | 10 131 | 187 | 652 |  |
| 600 | 8 923 | 189 | 705 |  |
| 1000 | 4 072 | 204 | 1210 |  |

1. Промежуточные слои модели.

Проведено тестирование по подпору промежуточных слоев L-Content. Для подбора промежуточного слоя модели зафиксируем значение шагов итерации:

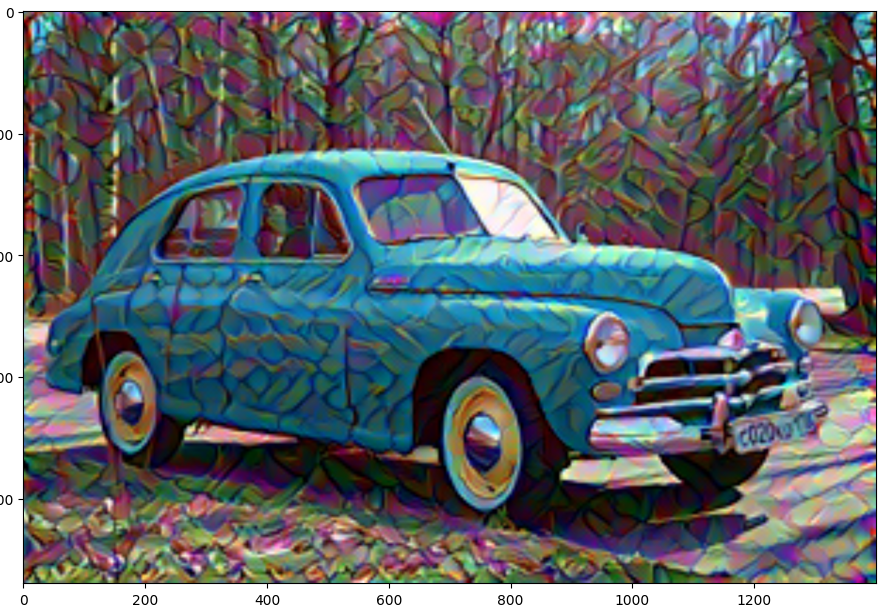
N step = 1000

list\_content = ['conv\_1', 'conv\_3','conv\_5']

list\_style = ['conv\_1', 'conv\_2', 'conv\_3', 'conv\_4', 'conv\_5']

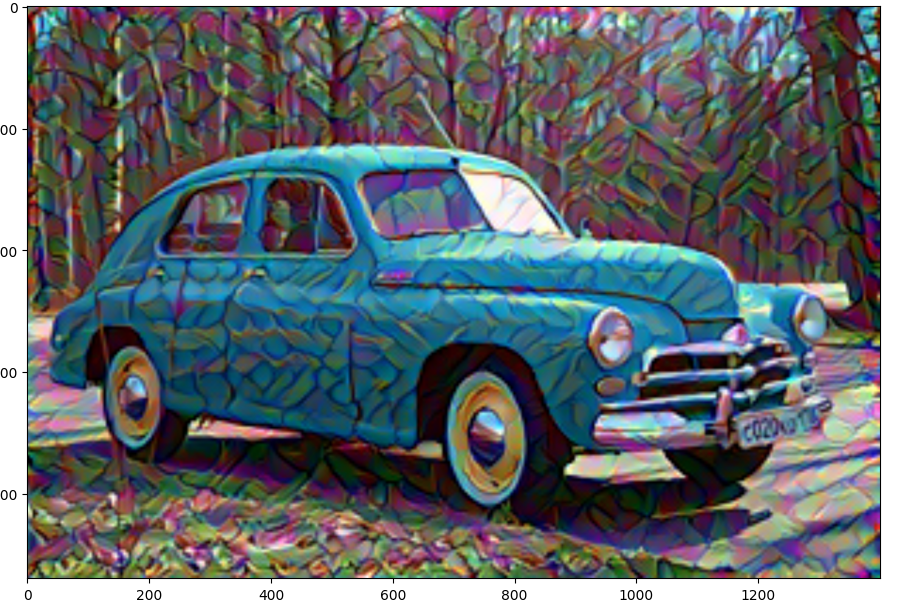
1. list\_content = 'conv\_1'

score\_style = 4 072 / score\_content = 204



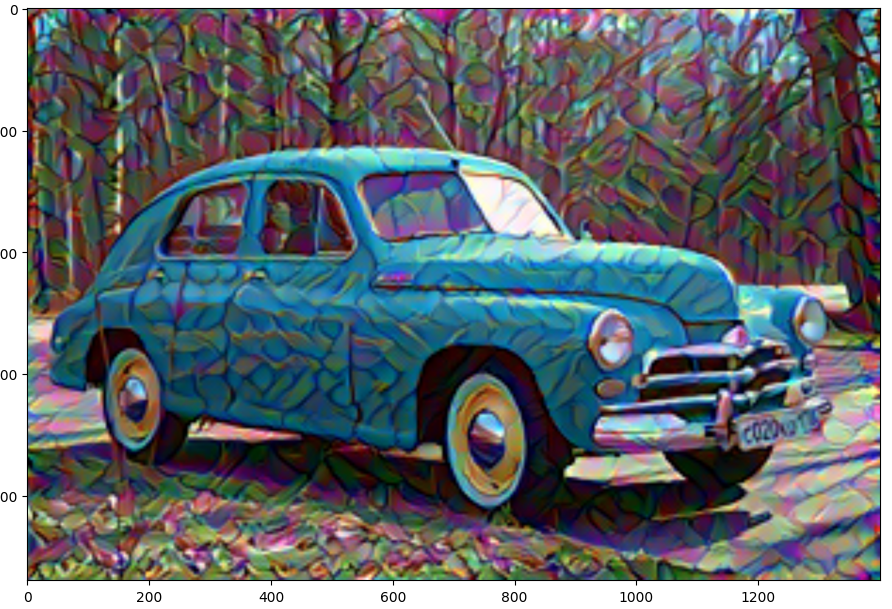
b) list\_content = 'conv\_3'

score\_style = 4 072 / score\_content = 204



1. list\_content = 'conv\_5'

score\_style = 4 072 / score\_content = 204



По результатам тестирования модели выявлено:

- с увеличением шагов итерации модели по переносу стиля, значение ошибки для стиля значительно падает.

- визуально контентное изображение приобретает характерные черты стиля.

- изменение промежуточного слоя L-Content - значимых отличий не выявлено.

- время обработки на CPU значительно увеличивается, с большим числом итераций.

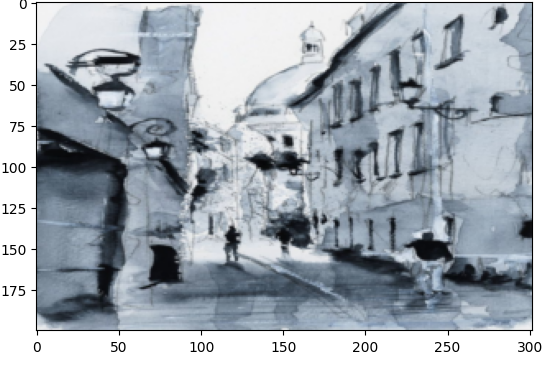
Принимаем гиперпараметры:

- промежуточные слои модели: для контента Content\_Layers = ['conv\_5'], для стиля Style\_Layers = ['conv\_1', 'conv\_2', 'conv\_3', 'conv\_4', 'conv\_5']

- количество шагов итерации, по умолчанию, для переноса стиля N\_Step = 600. А таже оставим пользователю возможность выбора глубины стилизации: минимальное значение 200, среднее значение 600, максимальное 1000.

Так же было проведено тестирование модели с применением графического ускорителя (GPU):

1. Изображение стиля:



1. Сгенерированное изображение:



Результат тестирования на GPU:

При значении N\_Step = 4000, значение ошибки стиля и контента максимально близки Loss\_Style = 17,6 Loss\_Content = 16,8. Визуально также заметно, что сгенерированное изображение очень близко по стилевому окрасу к исходному изображению стиля. Время работы модели на GPU в разы превосходит время работы на CPU и составило лишь 60 секунд.

Выводы

Задача стилизации изображения с применением методов глубокого обучения была успешно реализована. Простая архитектура модели VGG-19 и ее эффективность позволила создать приложение по переносу стиля. Использование небольших фильтров хорошо справляется с извлечением признаков на изображениях. А также, создано пользовательское приложение для стилизации изображений и фотографий с использованием мобильного устройства.

В ходе работы над проектом были выявлены направления для дальнейшего развития:

* Развертывание на удаленном сервере для непрерывной работы приложение.
* Повышение скорости обработки изображений, путем использования GPU.
* Добавление новых изображений стиля.
* Перенос кода Telegram-бота на асинхронный, для повышения скорости обработки команд.

Список литературы.

1. Архитектура модели VGG-19:

https://translated.turbopages.org/proxy\_u/en-ru.ru.97549cb4-67957399-c2fbf063-74722d776562/https/www.geeksforgeeks.org/vgg-net-architecture-explained/

1. Документация TelegramBotAPI:

https://pytba.readthedocs.io/ru/latest/index.html

1. Стили и направления в изобразительном искусстве

https://kalachevaschool.ru/blog/stili-i-napravleniya-v-iskusstve

https://vertex-art.ru/articles/stili-i-napravleniya-zhivopisi

1. «Простой Python» Б. Любанович.
2. «Изучаем Python» Э. Мэтиз.
3. Стилизация изображений с помощью нейронных сетей

https://habr.com/ru/companies/vk/articles/306916/