

**Deep Learning (семестр 1, осень 2021): продвинутый поток**

**Отчет  
по итоговому проекту  
“Telegram-боты. Перенос стиля в телеграм-ботах”**

**Виталий Бочаров  
stepik\_id\_56089805**

**2022 г.**

**Хочу выразить благодарность организаторам курса за предоставленную возможность погружения в мир Deep Learning.**

Виталий Бочаров  
12.02.2022

Итоговый проект выполнен по теме “Telegram-боты” в соответствии с техническим заданием (Приложение 1.) по Варианту 1. “Перенос стиля в телеграм-ботах”.

В ходе выполнения итогового проекта реализованы два варианта моделей переноса стиля с одной фотографии на другую, также сделан телеграм-бот, который может принимать от пользователя два изображения и возвращать в ответ изображение с перенесенным стилем.

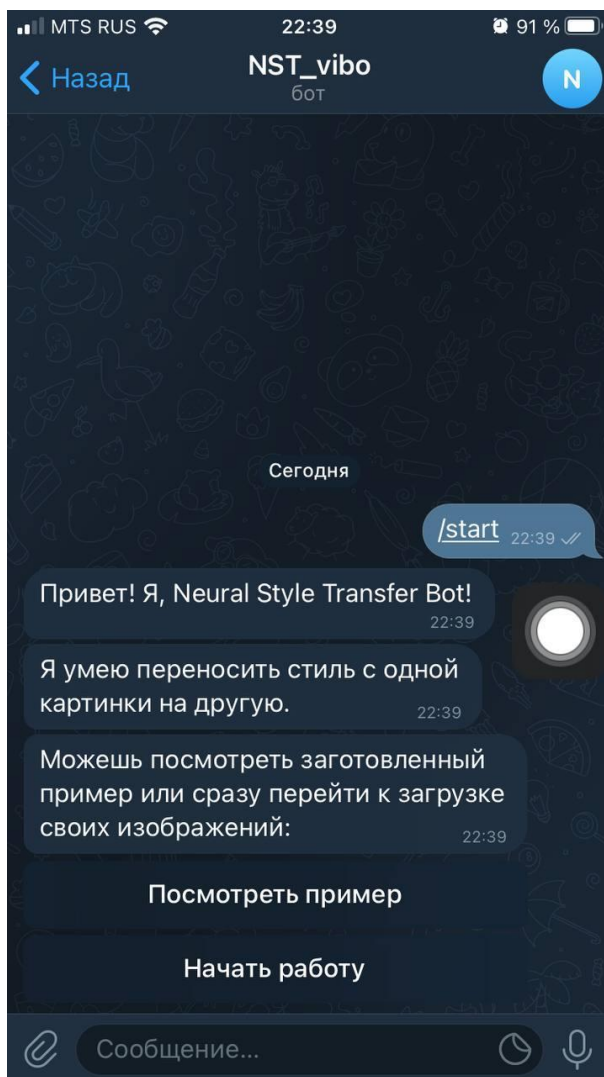
На подготовительном этапе работы были рассмотрены различные варианты Neural Style Transfer моделей (далее NST-модели) и фреймворков, в т.ч:

- Pytorch tutorials [https://pytorch.org/tutorials/advanced/neural\\_style\\_tutorial.html](https://pytorch.org/tutorials/advanced/neural_style_tutorial.html);
- “Медленный алгоритм” с семинара DLS от 11.01.2019  
[https://www.youtube.com/watch?v=u2HDm7YSwoA](https://www.youtube.com/watch?v=u2HDm7YSwoA;);
- Pystiche <https://github.com/pystiche/pystiche>;
- Pystiche: A Framework for Neural Style Transfer  
<https://medium.com/pytorch/pystiche-a-framework-for-neural-style-transfer-1ea6e4825f32>;
- NST in Pytorch <https://nextjournal.com/gkoehler/pytorch-neural-style-transfer>;
- All you need for Photorealistic Style Transfer in PyTorch  
<https://kushaj.medium.com/all-you-need-for-photorealistic-style-transfer-in-pytorch-acb099667fc8>.

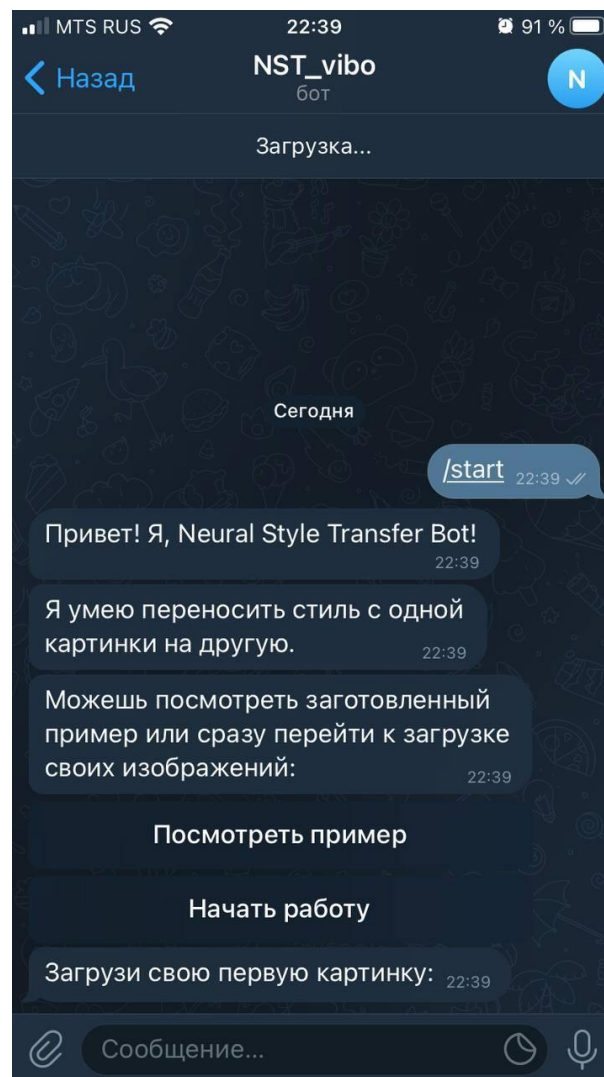
Для дальнейшего использования в итоговом проекте была выбрана модель из обучающего семинара DLS от 11.01.2019, а также для, сравнения, готовый фреймворк Pystiche.

В качестве API для телеграм-бота был использован telebot (<https://github.com/eternnoir/pyTelegramBotAPI>), опробован функционал по получению и сохранению данных от пользователей, диалоговых меню в чате и ответных сообщений от бота пользователю.

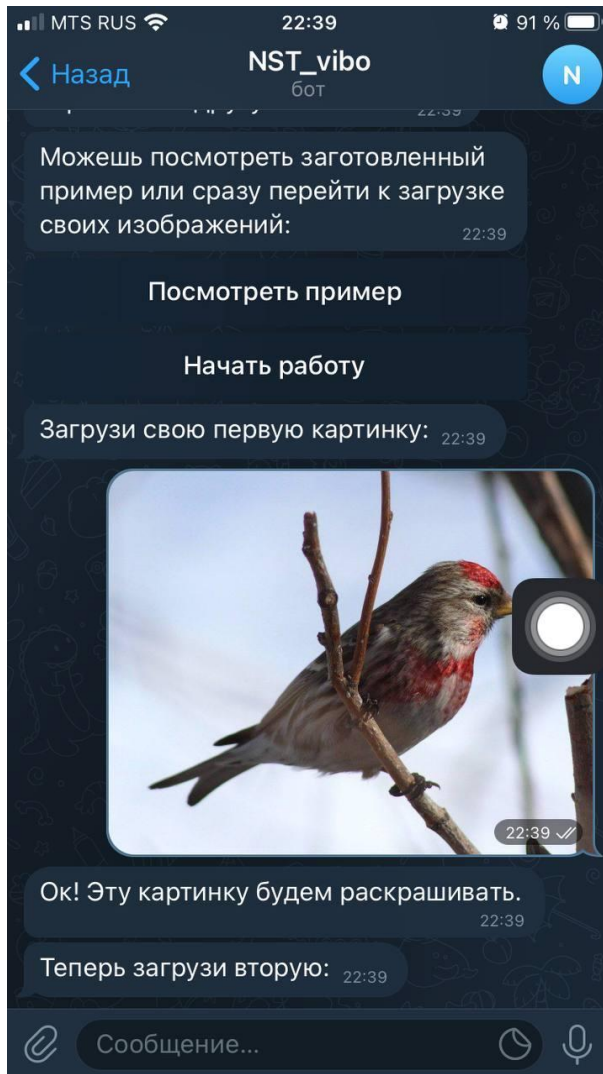
Пример работы телеграм-бота при запуске итогового ноутбука с проектом в среде GPU Google Colab и выбора пользователем модели из обучающего семинара DLS от 11.01.2019, приведен ниже.



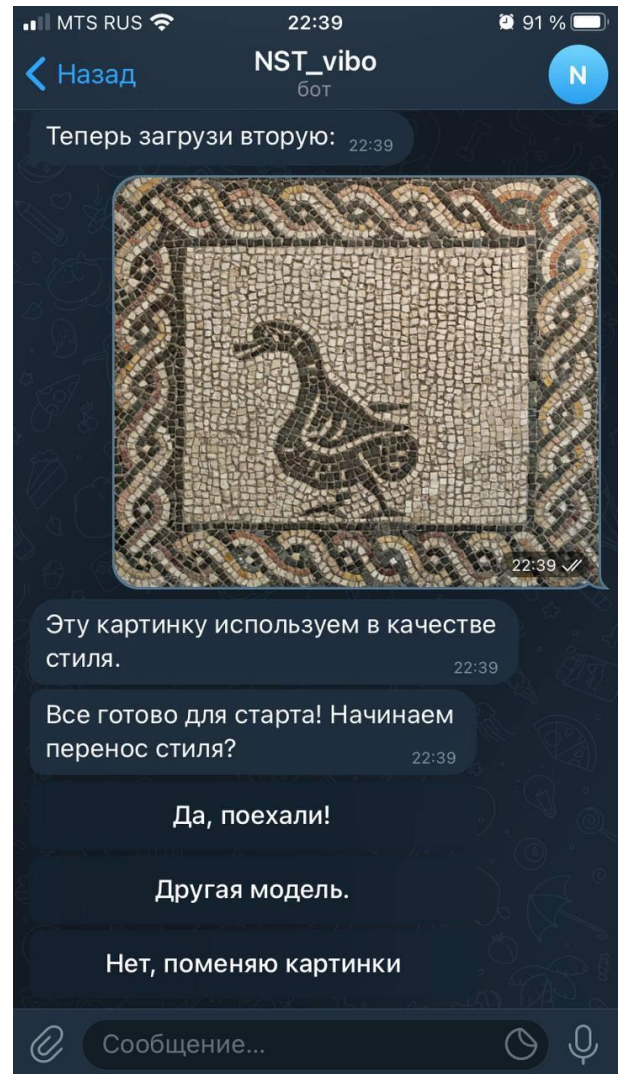
Шаг 1. После перехода по ссылке [https://t.me/NST\\_vibo\\_bot](https://t.me/NST_vibo_bot) и нажатия кнопки “Начать” пользователь попадает в приветствие, где реализовано меню выбора примера работы бота/начала работы пользователя с собственными изображениями. Пользователь выбрал “Начать работу”. Переход к шагу 2.



Шаг 2. Начало работы. Пользователю предлагается выбрать первое изображение, которое будет использовано как content\_image.

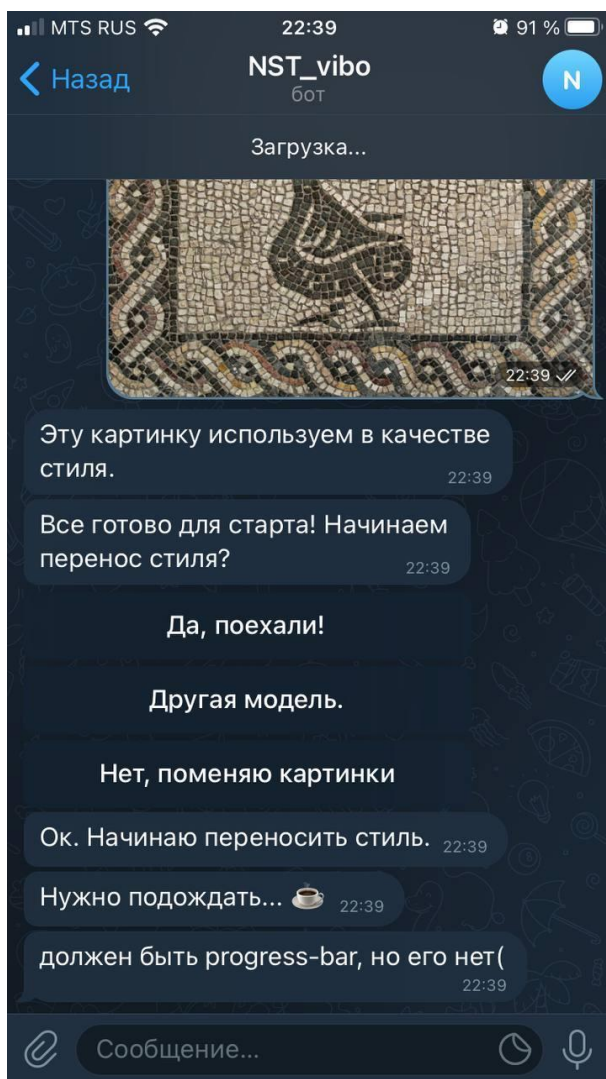


Шаг 3. После загрузки первого изображения пользователю предлагается загрузка второго изображения, которое будет использовано как `style_image`.

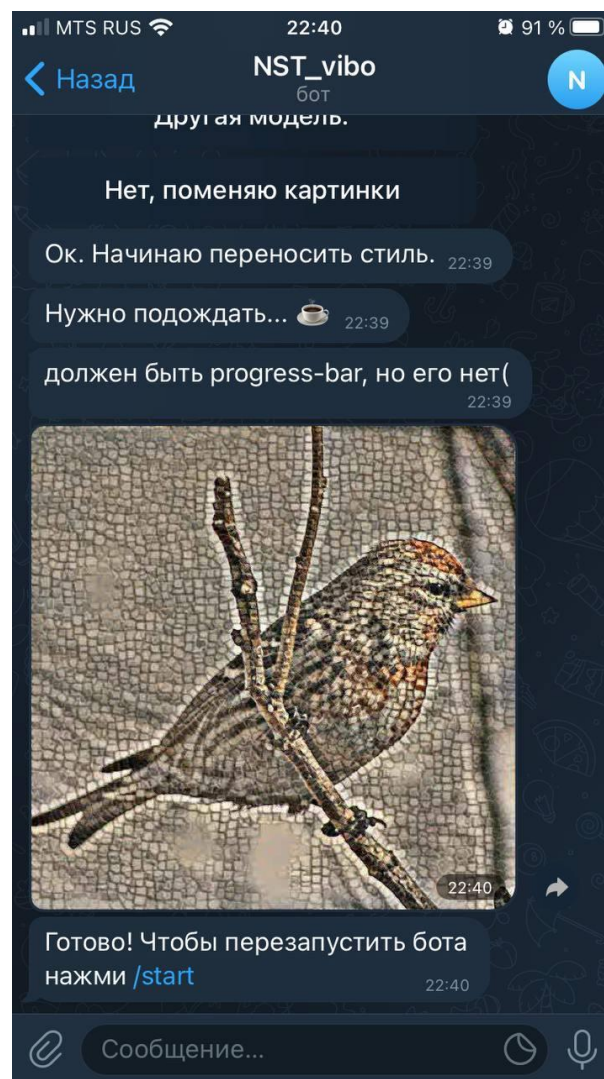


Шаг 4. После загрузки второго изображения пользователю предлагается меню выбора двух моделей / замены ранее выбранных изображений. Пользователь выбрал "Другая модель". Переход к шагу 5.





Шаг 5. Идет обработка изображения пользователя. Время ожидания при работе со стандартным GPU Google Colab составила порядка 2 минут. Прогресс-бар не реализован.



Шаг 6. Возврат пользователю полученного изображения (512\*512).

В качестве недостатков работы телеграм-бота стоит отметить чувствительность к вводу пользователем не предусмотренной моделью текстовой информации, а также отсутствие прогресс бара для информирования пользователя о времени ожидания.

Ниже приведены несколько примеров переноса стиля изображения телеграм-ботом при запуске итогового ноутбука с проектом в среде GPU Google Colab для двух моделей - Pystiche и модели из обучающего семинара DLS от 11.01.2019 (медленный алгоритм). Выходное изображение имеет размер 512\*512. Также моделями предусмотрена работа в среде CPU Goolge Colab с переходом к выходному изображению 256\*256.

Пример 1.

**Input\_image:**



Content\_image



Style\_image

Запринтим изображение, чтобы увидеть, как оно выглядит с точки зрения компьютера:

```
[[[111 138 227] ... [ 76 65 178]
 [110 132 225] ... [ 82 73 186]]]
```

А теперь, с помощью matplotlib (так удобнее) отрисуем изображение уже в виде изображения:

**Output\_image (GPU Google Colab, 512\*512):**



Pystiche. Время работы порядка 5 минут.

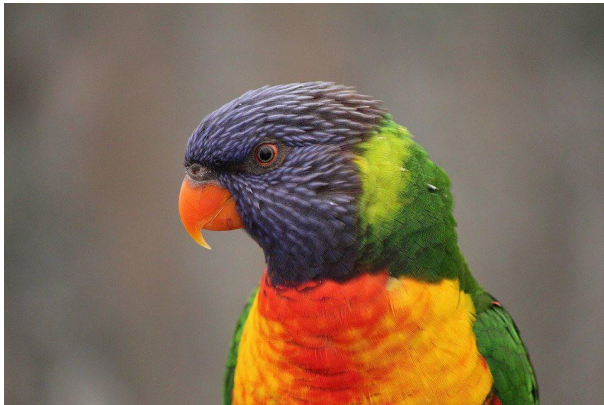


Медленный алгоритм. Время работы порядка 2 минут.



Пример 2.

**Input\_image:**

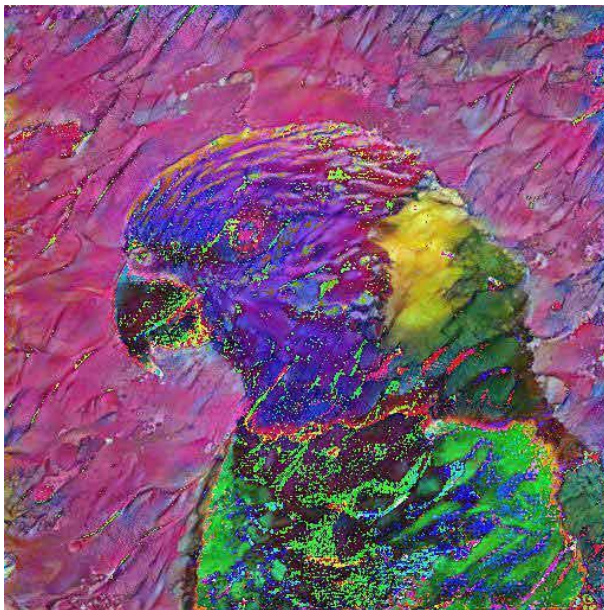


Content\_image



Style\_image

**Output\_image (GPU Google Colab, 512\*512):**



Pystiche. Время работы порядка 5 минут.



Медленный алгоритм. Время работы порядка 2 минут.



Пример 3.

**Input\_image:**



Content\_image



Style\_image

**Output\_image (GPU Google Colab, 512\*512):**



Pystiche. Время работы порядка 5 минут.



Медленный алгоритм. Время работы порядка 2 минут.



Пример 4.

**Input\_image:**

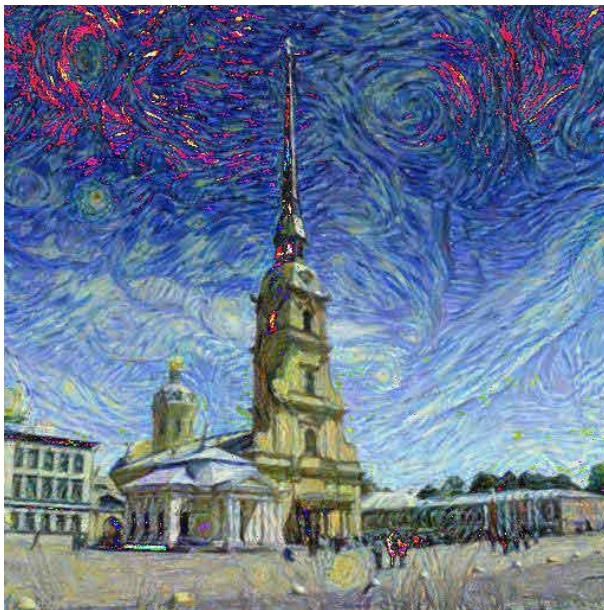


Content\_image

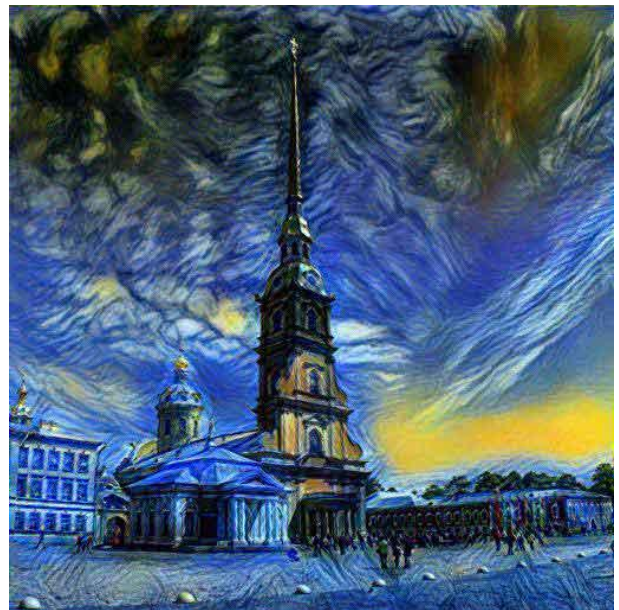


Style\_image

**Output\_image (GPU Google Colab, 512\*512):**



Pystiche. Время работы порядка 5 минут.



Медленный алгоритм. Время работы порядка 2 минут.

**Дальнейшее развитие проекта.**

На данном этапе не реализованным остался deploy бота на сервисе. Также не реализована асинхронность, из дополнительной части задания не добавлена возможность переноса стиля с помощью GAN'ов.

**Отчет для Stepik в запрашиваемом формате приведен в Приложении 2.**

**Приложения:**

1. Техническое задание (ссылка на Google);
2. Отчет на Stepik



Приложение 1.  
Техническое задание (ссылка на Google)

**Техническое задание**

<https://docs.google.com/spreadsheets/d/1900ETbH8zkbogVIDHWQGzIIN2Quy-R6QL0fLbfFai5Y/edit#gid=0>

**Поток**

Базовый и продвинутый

**Название**

Telegram-боты

**Ведёт**

Аркадий Боков

**Про что проект**

Стилизация изображений в телеграм-боте

**Что нужно будет сделать**

Основная часть (один из двух вариантов):

Вариант 1:

- Написать код простой модели, которая сможет переносить стиль с одной фотографии на другую. Можно использовать медленный алгоритм, который мы рассказывали на нашем занятии.
- Сделать бота, которому можно отправить две фотографии и получить в ответ фото с перенесенным стилем.

Дополнительная часть для варианта 1:

- Добавить возможность переносить стиль с помощью GAN'ов.

Вариант 2:

- Написать код, который улучшает качество фотографии с помощью GAN'ов (можно сужать выборку до каких-то однотипных объектов)
- Сделать бота, которому можно отправить фотографию и получить в ответ фото с улучшенным качеством

Дополнительная часть для варианта 2:

- Добавить возможность улучшать качество фотографии разных объектов (т.е. не однотипных)

**Основные этапы**

Для основной части:

- 1) Еще раз посмотреть на код в занятии и перенести всю функциональность в класс, чтобы потом можно было удобнее работать.
- 2) Написать бота, у которого будет диалог, предлагающий прислать картинки, и хэндлер, вызывающий класс из предыдущего пункта, чтобы сгенерировать ответную картинку.
- 3) (опционально) Асинхронность. Если вы раньше не писали приложения для сервера, то ваш код получится не асинхронным, а именно, во время обработки картинки бот будет просто висеть и не сможет ответить другому человеку. Это можно исправить, но потребуются хорошее понимание фреймворка, на котором Вы пишете бота.

Дополнительная часть:

- 1) Выбрать какой GAN Вы хотите использовать и какой стиль переносить (или для каких фотографий улучшать качество). Возможно, Вы захотите сделать не совсем перенос стиля или super-res, а какую-то другую обработку картинок, что тоже принимается.
- 2) Прочитать, как алгоритм работает и понять.
- 3) Теперь есть два варианта:
  - Найти готовую реализацию и как-то вызывать ее из своего кода.
  - Натренировать свой GAN. Этот вариант скорее всего потребует вычислительных мощностей и времени, поэтому выбирайте его только если уверены в своих силах.
- 4) Дописать бота, чтобы он мог использовать GAN.

Приложение 2.  
Отчет на Stepik

1. <https://drive.google.com/file/d/1EMwVepeOdK6J4hPAzhxuJ-40xMqAiYM8/view?usp=sharing>
2. [https://github.com/vi-bo/nst\\_bot](https://github.com/vi-bo/nst_bot)
3. [https://t.me/NST\\_vibo\\_bot](https://t.me/NST_vibo_bot)
4. telegram(@VitaliyBocharov), dls-community.slack.com(@vibo)
5. БОТ НЕ ЗАПУЩЕН. Вариант I. (перенос стиля). Для запуска в ноутбуке "v\_[project]telegram\_bot.ipynb" выполнить раздел "ИТОГОВЫЙ ПРОЕКТ".