

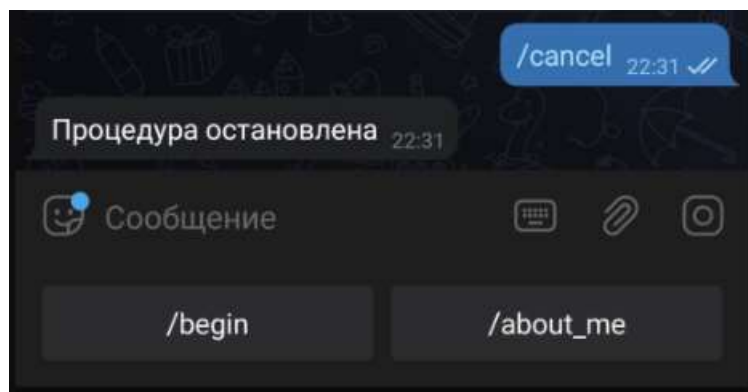
Отчет о создании телеграм-бота

DSL базовый поток осень 2021

Часть 1: бот

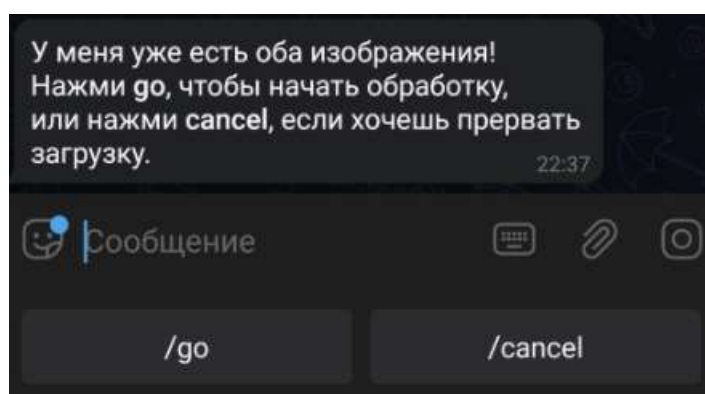
Данный бот был создан с использованием библиотеки `pyTelegramBotAPI`. В его основу легли как образцы кода, доступные в официальном репозитории самой библиотеки, так и образцы кода, взятые из различных источников, таких как обучающие видео на YouTube и различные статьи о создании чат-ботов, найденные на просторах глобальной сети не без помощи `google.com`.

Т.к. создание бота осуществлялось мною впервые, я столкнулся с немалым количеством проблем, многие из которых удалось решить, однако часть проблем осталась нерешенной - в частности, так и не удалось настроить `webhook` и бот в данный момент работает на `heroku` через пулинг в постоянном режиме `worker`, не засыпая. Также не удалось реализовать надписи на кнопках в чате на русском языке, на данный момент кнопки выглядят следующим образом:



С другой стороны реализация кнопочного интерфейса более-менее удалась, также удалось построить не только базовую логику взаимодействия с пользователем, но и закрыть большую часть ошибочных действий и возможных отступлений от основного маршрута, например, если послать третье изображение после того,

как контент и стиль уже приняты, то бот выдаст сообщение:



Наибольшее количество трудностей вызвал деплой бота на сервер - изначально мною был выбран сервис "яндекс-функция" и бот работал нормально через webhook, однако для данной функции установка библиотеки torch была невозможна и пришлось все же воспользоваться хостингом heroku. Каково же было мое удивление, когда нормально работающий на Яндексе бот на heroku падал без какой-либо причины и объявления войны. Узнать у heroku, в чем дело, так и не получилось и это вынудило меня вернуться к идее пулинга, после чего бота не без труда все же удалось вернуть в мир живых. На данный момент было создано и загружено более двадцати версий бота.

Часть 2: модель

Модель была взята из семинара DSL, посвященного переносу стиля и GAN и модифицирована для дальнейшей работы на хостинге. Самая главная трудность, с которой пришлось столкнуться, это лимит дискового пространства, из-за чего не удавалось даже скачать предобученную VGG19, хотя у сокурсников этот этап заканчивался благополучно. Для решения этой проблемы я загрузил VGG через ноутбук в коллабе и, обрезав модель и оставив только сверточные слои, сохранил ее в отдельном файле, а затем просто загружал в коде этот файл вместо скачивания готовой нейросети:

```
cnn = torch.load('new_model.pth')
```

Далее немного поигрался с параметрами, оптимизаторами и добился оптимальных настроек для более-менее приемлемой работы в условиях сильно ограниченного вычислительного ресурса.

Часть 3: итоги работы

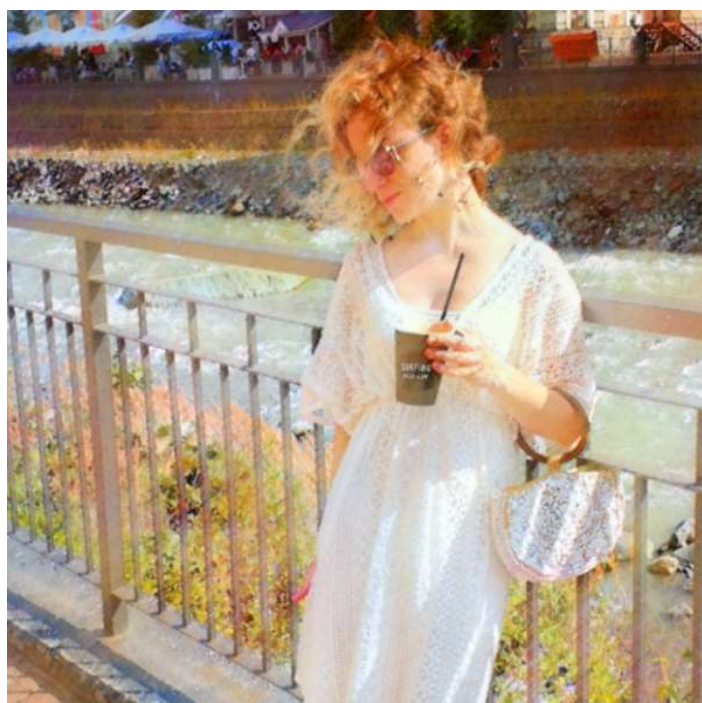
Итогом моей работы стал прежде всего функционирующий бот, способный предоставить хоть и несложный, но все же, на мой взгляд, понятный для пользователя интерфейс, а также модель, способная уместиться и работать на крайне скромном хостинге и исправно выдавать пусть и скромный, но все же стабильный результат.

Один из примеров работы бота:





Итоговое изображение:



Конечно, данная работа - это по сути самый минимум, необходимый для зачета, но я горд и этим, т.к. столкнулся с такими задачами

впервые в жизни. Мне действительно было очень приятно после стольких дней напряженной работы получить что-то работающее, то, что можно использовать по настоящему, что можно показать другим людям не без определенной гордости за результат, хотя много раз хотелось сдаться и бросить все на половине пути.

Часть 4: контактные данные

Проект был выполнен Поверенновым Александром, степик id 399257136.

Мои контакты:

Телеграм: <https://t.me/lunarycat>

ВК: <https://vk.com/lunarycat>

GitHub: <https://github.com/AlexanderPoverennov>