Областная научно-исследовательской конференции обучающихся

«Кузбасская школьная академия наук»

Наименование секции Научно-техническое творчество

Проектная работа

**Нейро-стилизация изображений**

Яковлев Павел,

учащийся 10 класса

НМБОУ «Гимназия №11», Анжеро-Судженский городской округ (муниципальный район)

Научный руководитель:

Мангазеев А.Ю.

учитель первой квалификационной категории,

программист, тиктокер, стример, пасечник, студент

НМБОУ «Гимназия №11» г. Анжеро-Судженск

Содержание

Введение…………………………………………………………3

1. История нейронных сетей…………………………………...4

2. Применение нейронных сетей………………………………6

3. Среда работы и необходимые библиотеки…….…...………8

4. Этапы разработки проекта………...….….….….….….…….9

5. Результаты преобразований бота…………………………..11

Заключение…………………………………………..…………12

Список использованной литературы…………..……………..13

Введение

Искусственный интеллект – наука и технология создания интеллектуальных машин, особенно интеллектуальных компьютерных программ, способные выполнять творческие функции, которые традиционно считаются прерогативой человека. Искусственный интеллект давно вышел за рамки технологий далекого будущего и находит широкое применение в настоящее время во всех сферах жизни общества. В настоящее время развлечения и социальные сети являются неотъемлемой частью нашего досуга. Поэтому данный проект будет довольно востребован в современном обществе.

***Цель***

Создать Telegram бота по преобразованию изображений с использованием генеративно-состязательных нейронных сетей

***Задачи:***

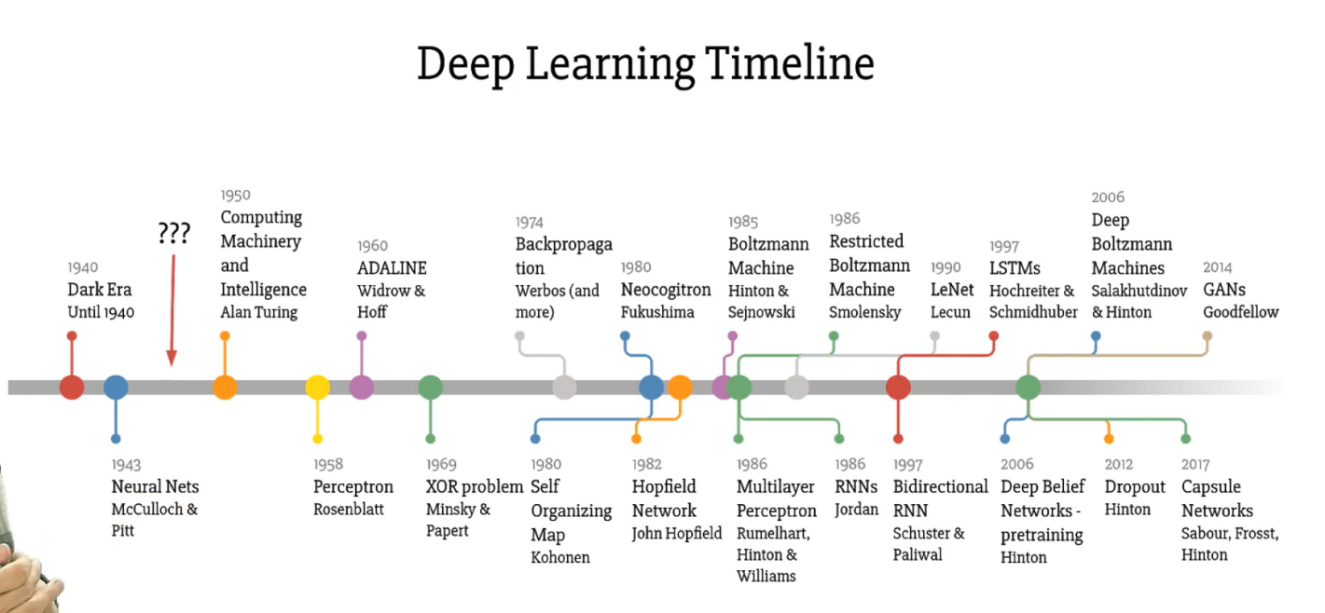
* Изучить синтаксис библиотеки PyTorch
* Изучить и реализовать архитектуру CycleGAN
* Выбрать задачу подходящую под выполнение данной архитектурой
* Найти, собрать и подготовить данные для выполнения поставленных задач
* Обучить модели для требуемых задач
* Адаптировать архитектуру для исполнения ботом
* Разместить Telegram-бота на сервере

*Обзор аналогов*

В данный момент аналогом данного проекта является Telegram бот Prisma, для обработки фотографий, но поддержка данного бота прекратилась либо приостановлена.

*Целевая аудитория*

Данный проект является полностью коммерческим и доступен абсолютно для всех пользователей мессенджера Telegram.

1. История нейронных сетей

Развитие искусственной нейросети началось с появлением электронно-вычислительных машин (ЭВМ) в конце 1940 года. В это время, канадский физиолог и нейропсихолог Дональд Хебб создал алгоритм нейронной сети, и заложил принципы его работы в ЭВМ. Затем, важными точками в развитии искусственных нейросетей были такие даты:

* **1954** год – на рабочей ЭВМ впервые на практике применили нейросеть.
* **1958** год – американским ученым по нейрофизиологии и искусственного интеллекта, Фрэнком Розенблаттом был разработан алгоритм распознавания образов и предоставлено его краткое изложение для общественности.
* **1960** год – ЭВМ не могла должным образом из-за слабых мощностей выполнять сложные задачи, возложенные на нее, поэтому интерес к технологии немного угас.
* За двадцатилетний период полным ходом шла «компьютеризация», и мощности тогдашних компьютеров хватило, чтобы вновь разжечь интерес к нейросетям. В 1980 году появилась система с механизмом обратной связи и начались разработки алгоритмов по самообучению.
* Спустя следующие 20 лет, мощности компьютеров выросли настолько, что в 2000 году ученые-исследователи смогли применять нейросети во многих сферах. Появились программы распознавания речи, имитация зрения, когнитивного восприятия информации. Нейросеть, машинное обучение, робототехника, компьютеризация стали частью нечто большего, под названием «искусственный интеллект».

2. Применение нейронных сетей

Исследовательские университеты, корпоративные гиганты, обучающие центры – применение нейросетей затронуло практически каждый аспект жизни человека.

*Поисковые системы*

[](http://future2day.ru/wp-content/uploads/2019/08/8.jpg)

Яндекс и Google не первый год используют нейронные сети для обучения собственный поисковых систем, делая их «умнее». Они адаптируются под конкретного пользователя, узнают о его предпочтениях и выдают максимально релевантные результаты. Тот же принцип применяется и в контекстной и таргетированной рекламе.

*Голосовые ассистенты*

[](http://future2day.ru/wp-content/uploads/2019/08/9.jpg)

Системы распознавания речи достигли такого уровня, что позволяют распознать вопрос и предоставить информацию по нему, или выполнить ту или иную функцию. Эти голосовые ассистенты внедряются в технику, поэтому мы можем голосом попросить включить медленную музыкальную композицию, притушить свет в комнате, открыть окно на проветривание и поставить таймер на приготовление еды в мультиварке.

*Автономное управление автомобилем*

[](http://future2day.ru/wp-content/uploads/2019/08/10.jpg)

Управление без участия водителя – это уже реальность сегодняшнего дня. Благодаря нейронной сети, что входит в комплекс систем автономного управления, автомобиль может передвигаться в автоматическом режиме, соблюдая все правила дорожного движения.

*Фотофильтры*

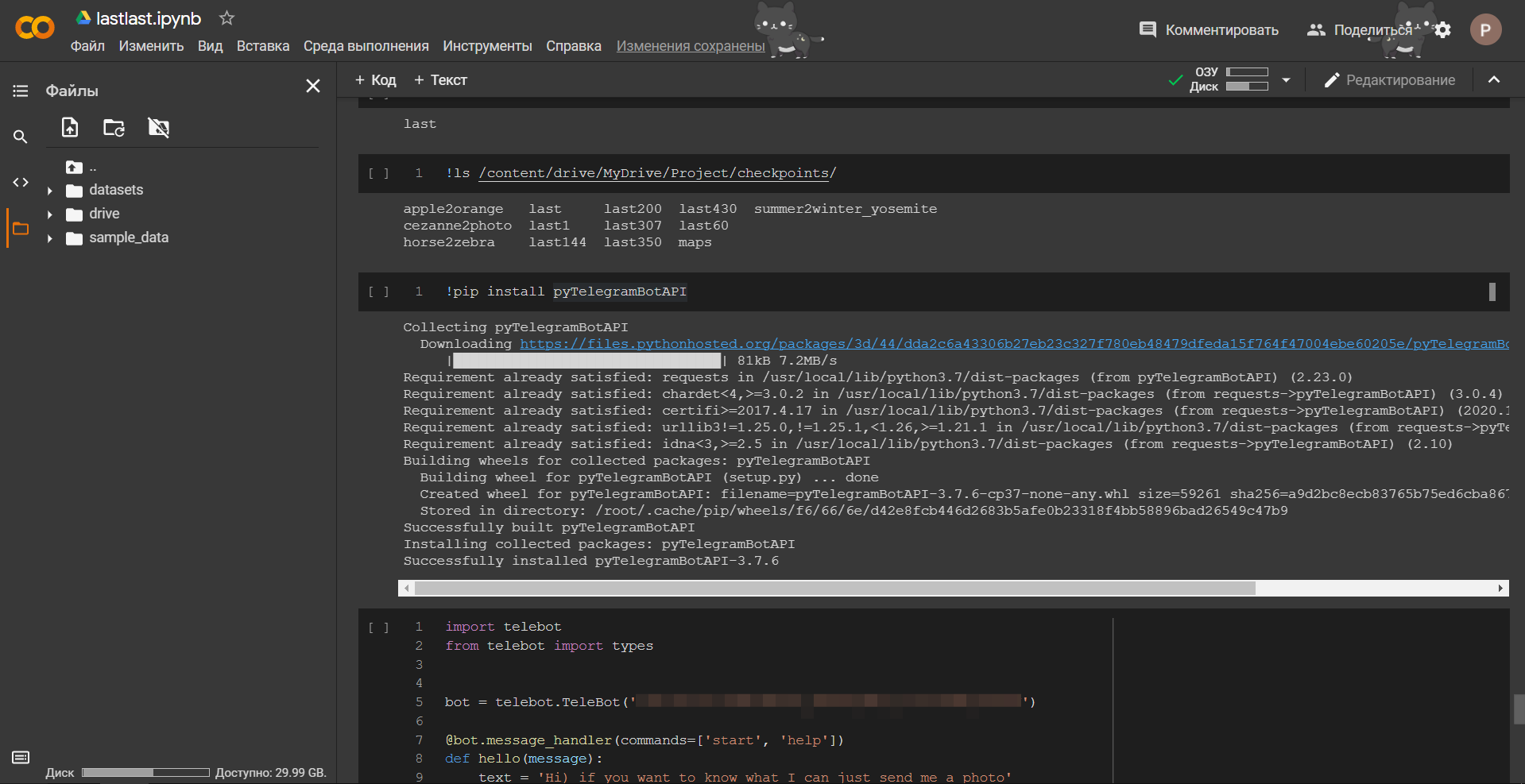
Работа с графикой – самое популярное направление нейросети. С ее помощью можно накладывать множество фильтров на фото и видео, используя дополнительную реальность. Еще, множество фотографий можно прогонять через специальные приложения, чтобы получить на выходе эффекты молодости, старения, смены пола и прочие прелести вполне качественного автоматического фотошопа.

3. Среда разработки и необходимые библиотеки

*Jupyter Notebook* – это крайне удобный инструмент для создания красивых аналитических отчетов, так как он позволяет хранить вместе код, изображения, комментарии, формулы и графики

[*PyTorch*](https://pytorch.org/) – это гибкий фреймворк для глубокого обучения, обеспечивающий автоматическое различение объектов при помощи динамических нейронных сетей. PyTorch поддерживает GPU-ускорение, [распределенное обучение](https://pytorch.org/docs/stable/distributed.html), различные виды [оптимизации](https://pytorch.org/blog/the-road-to-1_0/) и еще множество других приятных возможностей. Здесь я изложил некоторые мысли о том, как, на мой взгляд, следует использовать PyTorch; здесь не охвачены все аспекты библиотеки и рекомендуемые практики, но, надеюсь, этот текст окажется вам полезен.

[*Google Colaboratory*](https://colab.research.google.com/) — это облачный сервис, направленный на упрощение исследований в области машинного и глубокого обучения. Используя Colab, можно получить удаленный доступ к машине с подключенной видеокартой, причем совершенно бесплатно, что сильно упрощает жизнь, когда приходится делать глубокое обучение нейросетей.



4. Этапы разработки проекта

***Сбор данных***

Все данные взяты из отрытых источников: kaggle, google

Для выполнения поставленных задач понадобилось несколько датасетов: селфи-датасет (изображения с лицами людей), датасет с изображениями мультяшного героя, датасет с карандашным стилем портретного изображения, датасет с картами и изображениями со спутника.

***Построение архитектуры CycleGAN***

Для большинства архитектур преобразования изображения в изображение требуется большой набор попарных примеров, подготовить такой набор очень сложно и в некоторых задачах практически невозможно.

Поэтому в нашем проекте используется CycleGAN (циклическая генеративная состязательная сеть)

CycleGAN является дополнение архитектуры GAN, в ней задействованно сразу 2 генератора и 2 дискриминатора

Каждый генератор используется для непосредственного преобразования изображений, а дискриминаторы – для оценки их правдоподобности.

*Весь код в репозитории Github* [*https://github.com/PoulYak/StrangeFeaturesBot*](https://github.com/PoulYak/StrangeFeaturesBot)

***Обучение генеративно-состязательной сети***

Глубокие нейронные сети состоят из нескольких слоёв, каждый из них во время обучения способен захватывать определенную информацию с изображения в зависимости от того, на какой глубине находится слой и передавать оставшиеся данные следующему. Информация, извлекаемая глубокими слоями имеет решающее значение, в этих слоях содержится суть изображения вне зависимости от контекста

***Создание бота Telegram***

Для создания бота была использована библиотека pyTelegramBotApi

Создание бота именно в Telegram обосновано удобством работы с методами, предоставляемыми разработчиками Telegram и его стремительным ростом в популярности использования. Бот функционирует по технологии long pool, в перcпективе перейти на использование методов webhook.

***Размещение бота***

Для размещения бота было несколько вариантов платформ: Heroku, Google Cloud, AWS

При попытке размещения на некоторые из них возникли проблемы: Heroku сервисы Heroku не могут проводить такие вычисления, при попытке создания аккаунта на Google Cloud Platform, видимо, по законам РФ не разрешено создавать аккаунт для личного пользования.

Поэтому без лишних проблем бот был размещён на виртуальной машине AWS c EC2. Для передачи файлов с локального компьютера на VM (virtual machine) был использован крупнейший веб-сервис для хостинга IT-проектов GitHub.

5. Результаты преобразований бота

*Карандашный стиль*

*Мультяшный стиль*

*Преобразование изображений со спутника в карты*

Заключение

Удалось реализовать архитектуру, преобразовывающую изображения, обучить её нескольким стилям преобразований. Создать бота Telegram для удобного взаимодействия потребителей с самим продуктом.

В планах улучшать качество получаемых фотографий после преобразования и качество стилей, получаемых изображениями. Также в перспективе внедрить ещё стили и фичи, улучшить вычислительную мощность либо сменить размещение виртуальной машины также для увеличения производительности. Потенциальные потребители уже могут приступить к пользованию данным продуктом.

Ссылка на бота в Telegram <https://t.me/StrangeFeaturesBot>

Ссылка на репозиторий GitHub <https://github.com/PoulYak/StrangeFeaturesBot>

Личный сайт (скоро там появятся новые проекты) <https://poulyak.github.io/>

Список использованной литературы

1. CycleGAN [Электронный ресурс] <https://github.com/arnab39/cycleGAN-PyTorch> (дата обращения 15.01.2021).
2. Нейронные сети [Электронный ресурс] <https://future2day.ru/nejronnye-seti/> (дата обращения 28.02.2021).
3. Шпаргалка по TMUX [Электронный ресурс] <https://losst.ru/shpargalka-po-tmux> (дата обращения 14.02.2021).
4. Пишем ботов для Telegram на языке Python [Электронный ресурс] <https://groosha.gitbook.io/telegram-bot-lessons/> (дата обращения 01.02.2021).
5. How To Host Telegram Bot on AWS - Amazon Web Services [Электронный ресурс] <https://youtu.be/MApTRT37B0k> (дата обращения 05.02.2021).
6. Unpaired Image-to-Image Translation using Cycle-Consistent Adversarial Networks [Электронный ресурс] <https://youtu.be/m47qsfSZoTI> (дата обращения 14.01.2021).
7. Selfie dataset [Электронный ресурс] <https://www.crcv.ucf.edu/data/Selfie/>

(дата обращения 20.01.2021).