

Создание искусственного интеллекта на примере чат-бота идентификации покемонов

25.03.2019

—

Алиса

Обзор

1. Что такое искусственный интеллект

Чтобы дать определение понятию “искусственный интеллект”, необходимо в первую очередь определиться со значением самого слова “интеллект”. Согласно наиболее употребляемому определению, интеллект - способность системы в ходе самообучения создавать программы для решения задач определенного класса сложности и решать эти задачи. (Ильясов Ф. Н. Разум искусственный и естественный // Известия АН Туркменской ССР, серия общественных наук. 1986. № 6. С. 46—54.) Соответственно, искусственным интеллектом можно будет назвать техническую или программную систему, обладающую интеллектом и способную правильно интерпретировать внешние данные для достижения определенных целей. (Andreas Kaplan; Michael Haenlein (2019) Siri, Siri in my Hand, who's the Fairest in the Land? On the Interpretations, Illustrations and Implications of Artificial Intelligence, Business Horizons, 62(1))

2. Чем он полезен, для чего применяется

(<https://mentamore.com/covremennye-texnologii/oblasti-primeneniya-iskusstvennogo-intellekta.html>)

Преимущества искусственного интеллекта - способность автоматизировать процессы, ранее требовавшие необходимость человеческого участия и высокая скорость и точность анализа больших объемов информации. В сочетании с отсутствием подверженности человеческому фактору, эти свойства позволяют использовать ИИ практически во всех основных сферах человеческой жизни. В медицине искусственный интеллект применяется для диагностики различных заболеваний, анализа генетической предрасположенности к патологиям, некоторые системы способны моделировать лекарства. В промышленности машины могут осуществлять сбор деталей и бухгалтерские расчеты. Постоянно усложняющиеся и улучшающиеся алгоритмы позволяют искусственному интеллекту исполнять роль консультанта при общении с покупателями или анализировать резюме для повышения эффективности подбора персонала. Большую популярность набирают системы “умного дома” и роботы-пылесосы.

3. Проблема:

Несмотря на многочисленные достоинства, существует и ряд проблем, возникающих при использовании ИИ. Одно из очевидных препятствий - недостаток вычислительной мощности для поиска моделей и постройки алгоритма сравнения. Если рассматривать систему, распознающую объекты, цель моего проекта, то для реализации алгоритма необходимо либо использование готовых нейросетей, что в данном конкретном случае невозможно из-за того, что обучение как правило проводится на часто встречающихся моделях, таких как люди, транспортные средства, архитектурные строения.

ИИ для реализации алгоритмов требует либо а) большие вычислительные ресурсы для построения моделей поиска и сравнения, либо б) использование заранее обученных нейронных сетей, но доступные для использования заранее обученные сети, натренированы на наиболее встречающихся в реальной жизни объектах - образы людей, автомобили, дорожные знаки, животные, архитектурные объекты. При отсутствии мощных серверов возникают сложности с поиском и идентификацией нестандартных объектов.

4.

Цели

1. В данном проекте будет на практике проверена возможность создания искусственного интеллекта, способного распознавать образы, отсутствующие в стандартных базах моделей, используя вычислительные ресурсы бытового компьютера, на примере распознавания изображений покемонов.
2. Обеспечить возможность практического использования создаваемой нейросети людьми без специальных знаний и оборудования в виде бота для telegram.

Используемое техническое оборудование

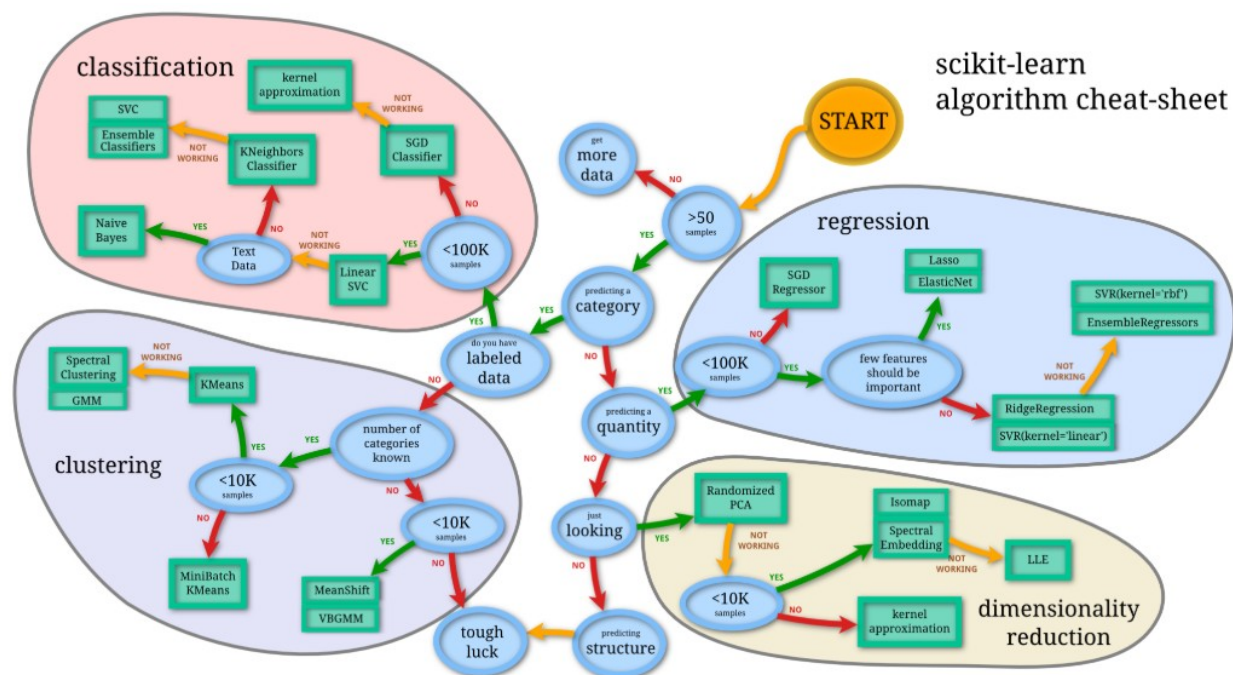
Одним из критериев для выбора алгоритма машинного обучения и параметров построения модели явилась вычислительная мощность компьютера, находящегося в распоряжении автора проекта:

1. Процессор Intel Core i5-8600K 3,6 ГГц.
2. Размер оперативной памяти 16 Гб.
3. Операционная система Windows 10, 64-разрядная.
4. Дискретная видеокарта - отсутствует.

Ключевые этапы

I. Выбор алгоритма и технических средств для создания ИИ

Путеводитель по популярным алгоритмам ИИ (за исключением нейронных сетей):



https://scikit-learn.org/stable/tutorial/machine_learning_map/index.html

1. Язык программирования для разработки - Python 3. Обосновать выбор
2. Выбор алгоритмов для использования в ИИ -
 - i. PCA https://en.wikipedia.org/wiki/Principal_component_analysis
 - ii. [SVM classifier](#)
3. Используемые библиотеки для распознавания, работы с файлами, изображениями, для работы с Telegram Bot API и т.д. Обосновать выбор

II. Установка программного обеспечения

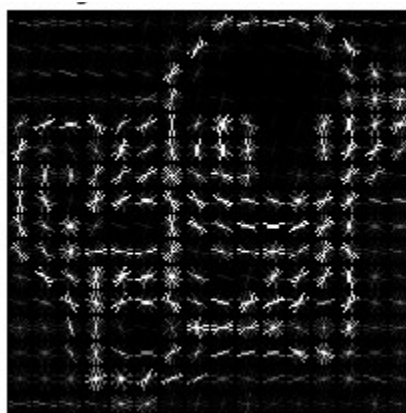
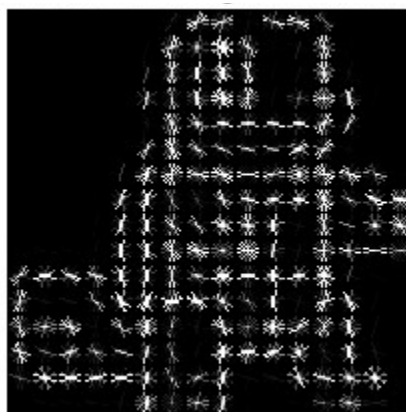
Введите свой текст здесь Введите свой текст здесь Введите свой текст здесь Введите свой текст здесь Введите свой текст здесь

III. Подготовка и обработка изображений для использования в качестве модели

Изображения приводятся к единому размеру, путем вписывания в квадрат.

Далее изображения обрабатываются методом HOG (Histogram of Oriented Gradients) - гистограмма направленных градиентов - изображение разбивается на блоки, сравнивается яркость соседних блоков и формируется массив векторов, показывающих направление и величину изменения яркости между блоками. Это позволяет выделить наиболее существенные части изображения - контуры где происходит изменение цвета, а более однородный фон убрать из изображения.

Результат работы алгоритма:



IV. Разработка алгоритма работы искусственного интеллекта

V. Обучение модели

Пример вывода на консоль при запуске программы teach.py:

```

Запуск обучения
2.66 сек занял сбор и преобразование образцов изображений
0.01 сек занял расчет количества главных компонент
Количество главных компонент векторов для 80% точности: 78
Идет процесс проецирования полных векторов изображений на оси главных компонент
0.0 сек занял расчет проекций главных компонент векторов
Найдены наилучшие параметры модели:
SVC(C=1, cache_size=200, class_weight='balanced', coef0=0.0,
    decision_function_shape='ovr', degree=3, gamma='auto_deprecated',
    kernel='rbf', max_iter=-1, probability=True, random_state=None,
    shrinking=True, tol=0.001, verbose=False)
0.32 сек заняло обучение модели

```

VI. Тестирование искусственного интеллекта

Одновременно с обучением модели осуществляется проверка качества ее работы на тестовых образцах, для чего из обучающих образцов отбирается 20% изображений, которые не используются для обучения, а используются только для пробного распознавания и по результатам такого распознавания вычисляется качество модели.

Пример вывода программы teach.py в процессе проверки модели:

```

Запуск тестирования качества распознавания на тестовых образцах
0.0 сек заняло тестирование модели
Результат тестирования качества модели:
    precision  recall  f1-score  support

Бульбазавр   1.00    1.00    1.00      2
    Генгар    1.00    0.25    0.40      4
Джигглипафф  1.00    0.17    0.29      6
    Дигглет   1.00    1.00    1.00      2
    Дитто     1.00    1.00    1.00      5

```


Пикачу	1.00	0.20	0.33	5
Сквиртл	0.80	0.80	0.80	5
Тангела	0.13	1.00	0.24	2
Чармандер	1.00	0.67	0.80	3
Черизард	0.86	0.86	0.86	7
<i>micro avg</i>	0.63	0.63	0.63	41
<i>macro avg</i>	0.88	0.69	0.67	41
<i>weighted avg</i>	0.91	0.63	0.65	41

```
[[2 0 0 0 0 0 0 0 0 0]
[0 1 0 0 0 0 0 3 0 0]
[0 0 1 0 0 0 0 4 0 1]
[0 0 0 2 0 0 0 0 0 0]
[0 0 0 0 5 0 0 0 0 0]
[0 0 0 0 0 1 0 4 0 0]
[0 0 0 0 0 0 4 1 0 0]
[0 0 0 0 0 0 0 2 0 0]
[0 0 0 0 0 0 1 0 2 0]
[0 0 0 0 0 0 0 1 0 6]]
```

Обучение завершено. Всего потребовалось 3.01 сек

Таким образом, взвешенная точность определения тестовых образцов составила 91%

VII. Разработка чат-бота

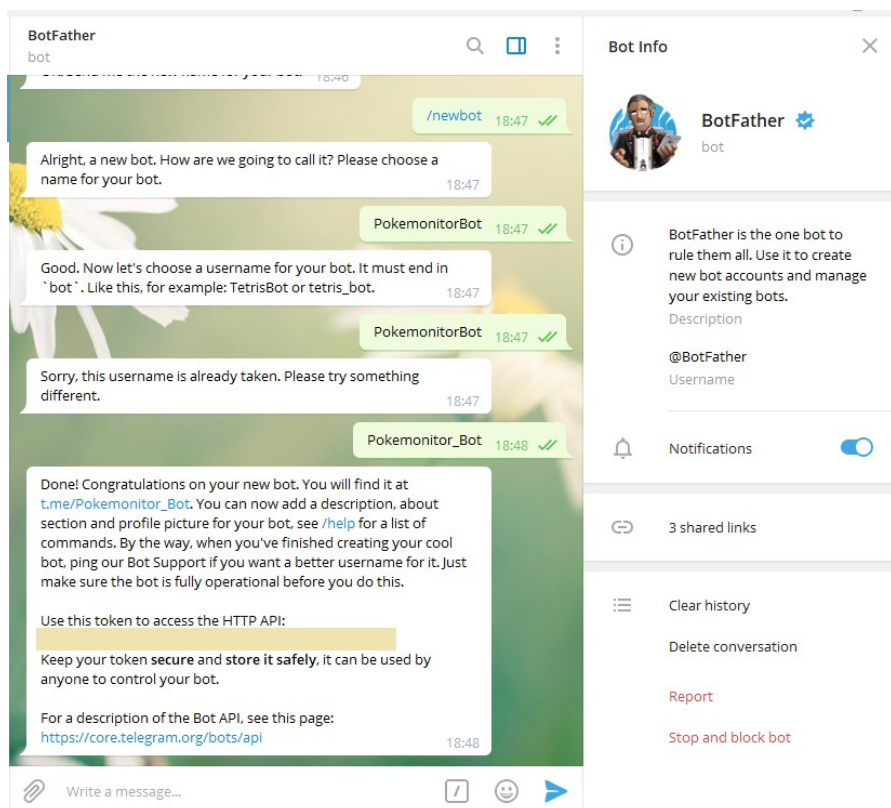
Для простого и свободного доступа к разрабатываемому искусственному интеллекту было решено не создавать собственный интернет сайт, а создать и подключить бота к популярному сервису обмена сообщениями Telegram. Telegram был выбран за простоту создания и легкость подключения бота.

Для обеспечения возможности подключения бота к Telegram, чтобы пользователи могли его находить и обмениваться сообщениями, был получен идентификационный ключ по инструкции на <https://core.telegram.org/bots#6-botfather>.

В Telegram боту @BotFather была дана команда /newbot для регистрации нового бота, а затем

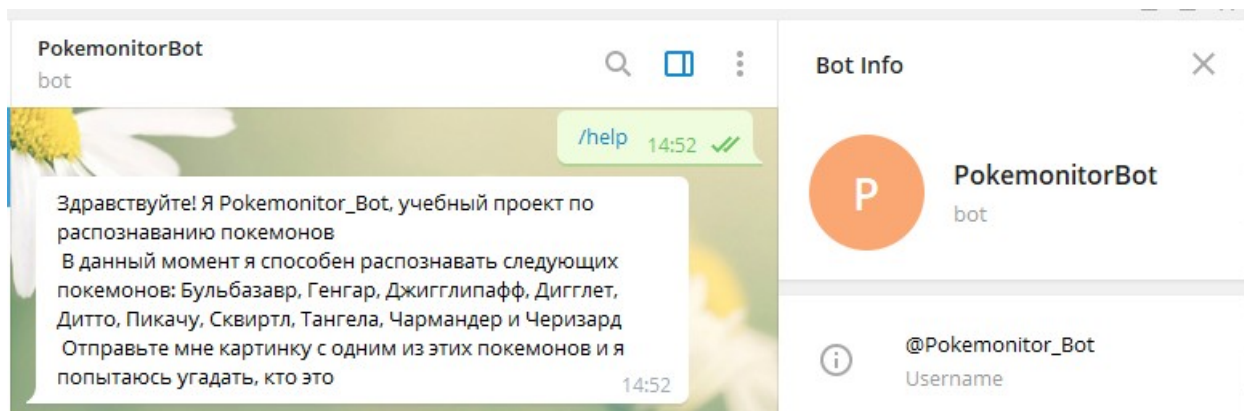
были придуманы имя и логин для своего бота.

Был зарегистрирован бот с именем Pokemonitor_Vot и получен секретный ключ для его аутентификации.



Для разработки программного кода бота использовалась бесплатная библиотека pyTelegramBotAPI (<https://github.com/eternnoir/pyTelegramBotAPI>).

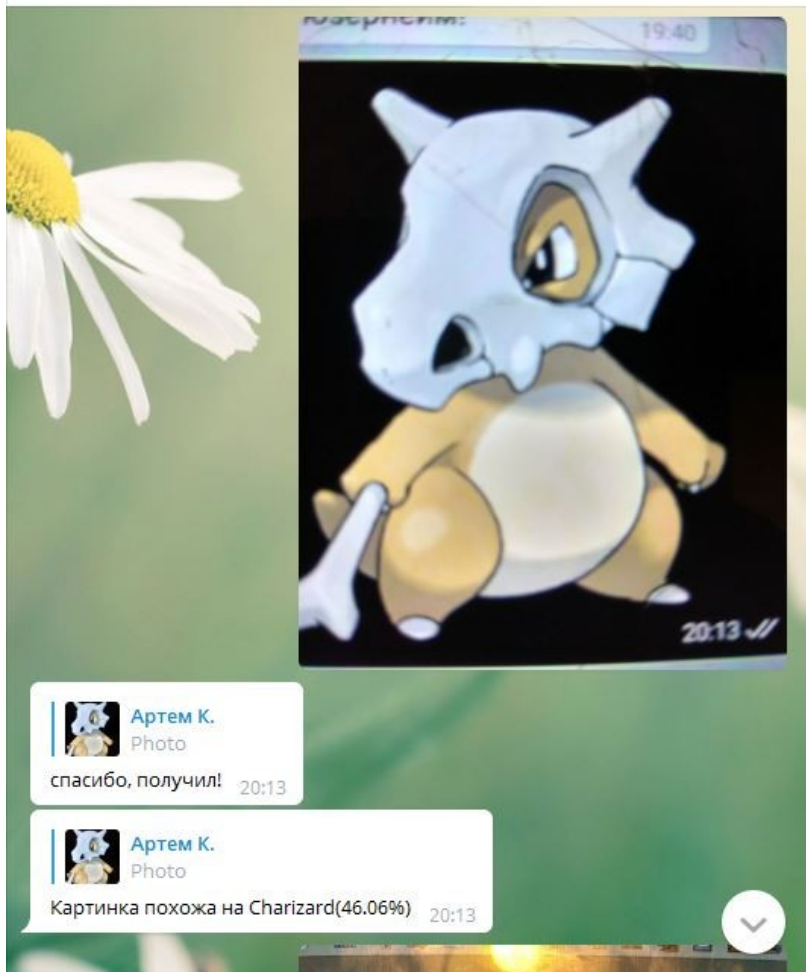
При начале общения пользователей с ботом, он представляется и сообщает порядок работы.



Результат работы бота с искусственным интеллектом:

PokemonitorBot
bot

19:40



20:13 ✓

Арте́м К.
Photo
спасибо, получил! 20:13

Арте́м К.
Photo
Картинка похожа на Charizard(46.06%) 20:13

Write a message...

Bot Info

PokemonitorBot
bot

@Pokemonitor_Bot
Username

Notifications ☒

10 photos

Add to Group

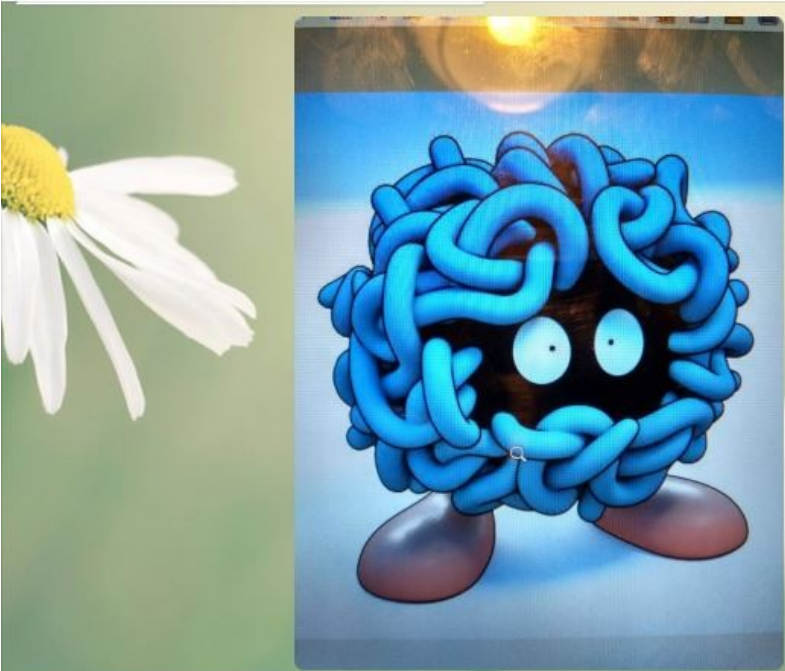
Clear history


Delete conversation

Report


Stop and block bot

PokemonitorBot
bot








 **Артем К.**
Photo


спасибо, получил! 20:15


 **Артем К.**
Photo



Картинка похожа на Pikachu(41.05%)
Tangela(42.64%) 20:15


 | Write a message...   


Bot Info 

 **PokemonitorBot**
bot

 @Pokemonitor_Bot
Username

 Notifications 

 10 photos

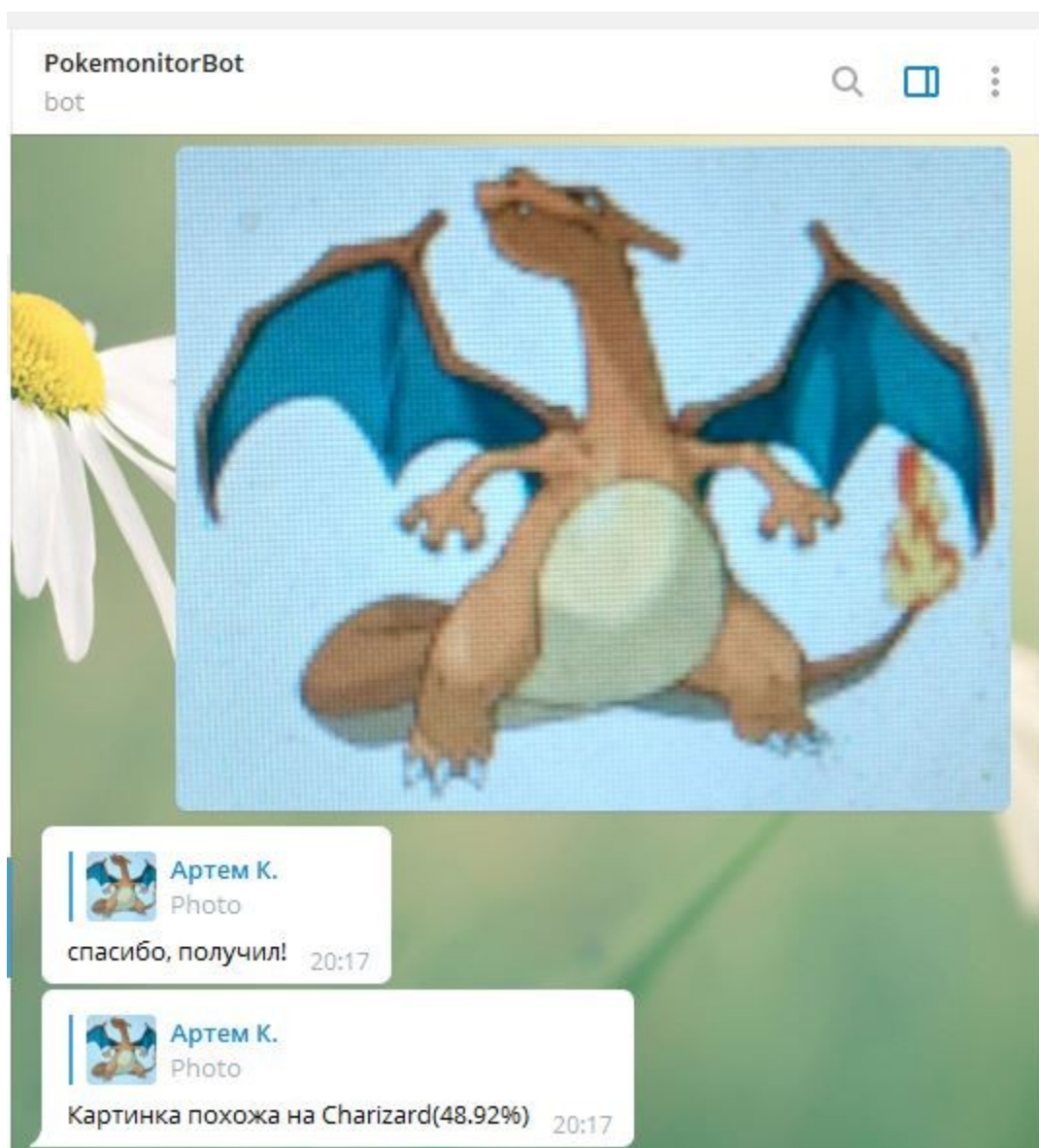
 Add to Group

Clear history

Delete conversation

Report

Stop and block bot





Результаты

Были разработаны следующие программные библиотеки:

- ❖ bot.py - непосредственно бот для Telegram, осуществляющий диалог с пользователем
- ❖ check_image.py - классифицирует полученное изображение
- ❖ image_modify - библиотека, приводящая изображения к единому размеру и формату, а также выявляющая на них контуры объектов и преобразующая их в векторное представление.
- ❖ process_images.py - подготавливает образцы обучающих изображений для тренировки модели
- ❖ teach.py - обучает модель ИИ классифицировать изображения на подготовленных образцах

Файловая система проекта:

- ❖ `trained_model` - папка, содержащая обученную модель и справочные таблицы
 - `pokemon_model.names` - справочник соответствия номера категории изображения и названия покемона
 - `pokemon_model.pca` - хранение параметров метода pca (метод главных компонент, сокращает число измерений вектора)
 - `pokemon_model.pkl` - хранения параметров метода svm, он же svc (метод опорных векторов, находит уравнения поверхностей, разделяющих категории)
- ❖ `pokemon` - необработанные изображения для обучения модели
- ❖ `pokemon_processed` - изображения, приведенные к единому виду и формату для обучения модели
- ❖ `bot_downloaded_images` - изображения, которые бот получает от пользователя

В ходе работы было осуществлено следующее:

1. Собраны и при помощи созданной программы `process_images.py` автоматически обработаны образцы изображений персонажей Pokemon для приведения их к единому размеру и типу, наиболее подходящему для обучения.
2. Создана программа `teach.py` для создания модели классификации изображений по собранным образцам на языке Python с использованием библиотеки `scikit-learn`. Изображения обрабатывались методом HOG (гистограмма направленных градиентов) для выявления основных контуров объектов и представлялись в виде многомерного вектора, который при помощи PCA (метода главных компонент) упрощался, оставляя только существенные для отбора размерности, и, используя SVC (классификацию методом опорных векторов), определялись параметры модели для использования при распознавании.
3. Используя программу `teach.py` создания модели, была обучена модель классификации покемонов на собранных образцах, качество работы которой было проверено на случайно отобранных классифицированных образцах.
4. Разработан алгоритм `check_image.py` классификации изображения по обученной модели. Подобно алгоритму обучения, алгоритм классификации обрабатывает проверяемое изображение, приводя к единому размеру и типу, преобразует его в многомерный вектор, уменьшает размерность вектора методом PCA, при помощи SVC находит степень схожести вектора с классами, имеющимися в обученной ранее модели.
5. Разработан алгоритм `bot.py` для получения изображений от пользователей через мессенджер Telegram, классификации их алгоритмом `check_image.py`, передачи пользователям ответного сообщения с указанием классов покемонов, наиболее схожих с полученным изображением.
6. Разработанный бот подключен к telegram и проведена проверка его работоспособности.

Выводы

Данная работа на практике показала, что разработка алгоритмов искусственного интеллекта доступна даже для школьников благодаря бесплатным и доступным программным средствам и библиотекам, реализующим необходимые математические функции. Обучение простой модели классификации и распознавания изображений занимает на обычном бытовом компьютере несколько минут, а распознавание новых изображений по уже обученной модели - доли секунды.

Разработанный бот для мессенджера Telegram, оснащенный искусственным интеллектом, способен классифицировать обработанные рисованные изображения персонажей комиксов, мультфильмов и игр Рокетоп с точностью до 91%, что достаточно для развлекательных и учебных целей.

При распознавании необработанных изображений, где изображение персонажа занимает менее 85% площади всего изображения, где изображено несколько персонажей или персонажи находятся в положении или позициях, отсутствующих в обучающих образцах, точность распознавания резко уменьшается до порядка 20-40%.

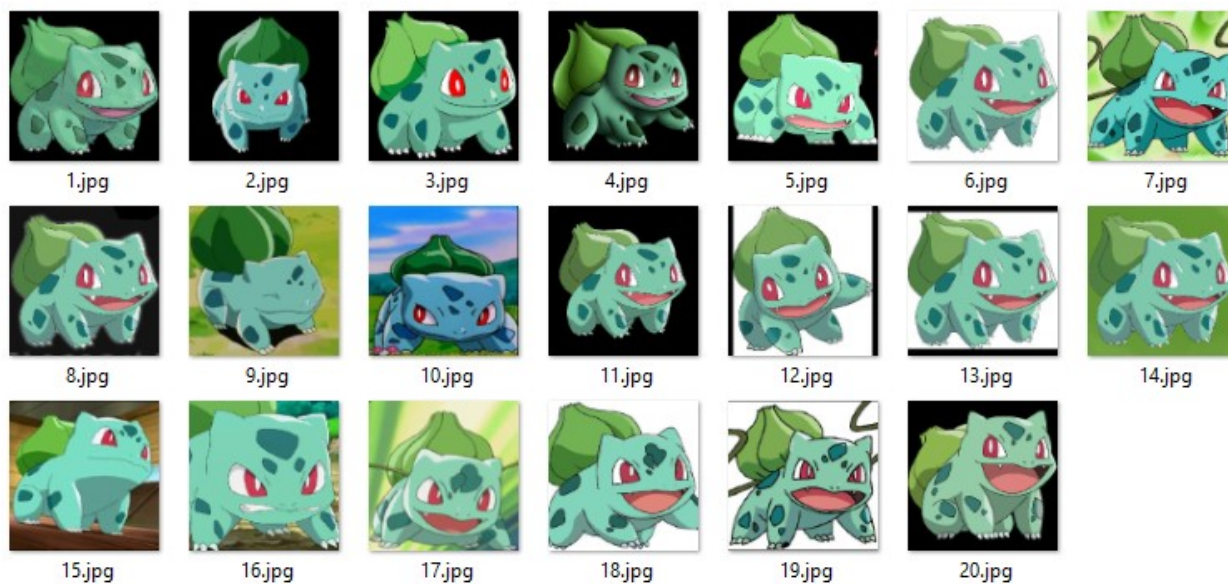
Дальнейшее развитие проекта возможно в следующих направлениях:

1. Улучшение качества распознавания образов:
 - Увеличение количества категорий распознаваемых образов (видов покемонов).
 - Увеличение количества обучающих образцов изображений для каждого вида покемонов.
 - Сравнение различных алгоритмов классификации и настраиваемых параметров их работы для выбора наилучшего из них.
 - Применение нейронных сетей в различных вариантах: готовых моделей на нейронных сетях или с дообучением готовых моделей собственными образцами.
2. Расширение возможностей:
 - Добавление способности самообучения бота на основе присылаемых ему изображений.
 - Вывод пользователям дополнительной справочной информации по распознанным покемонам.
 - Распознавание на видео встречающихся покемонов.
 - Поиск на видео кадров с заданным видом покемона.

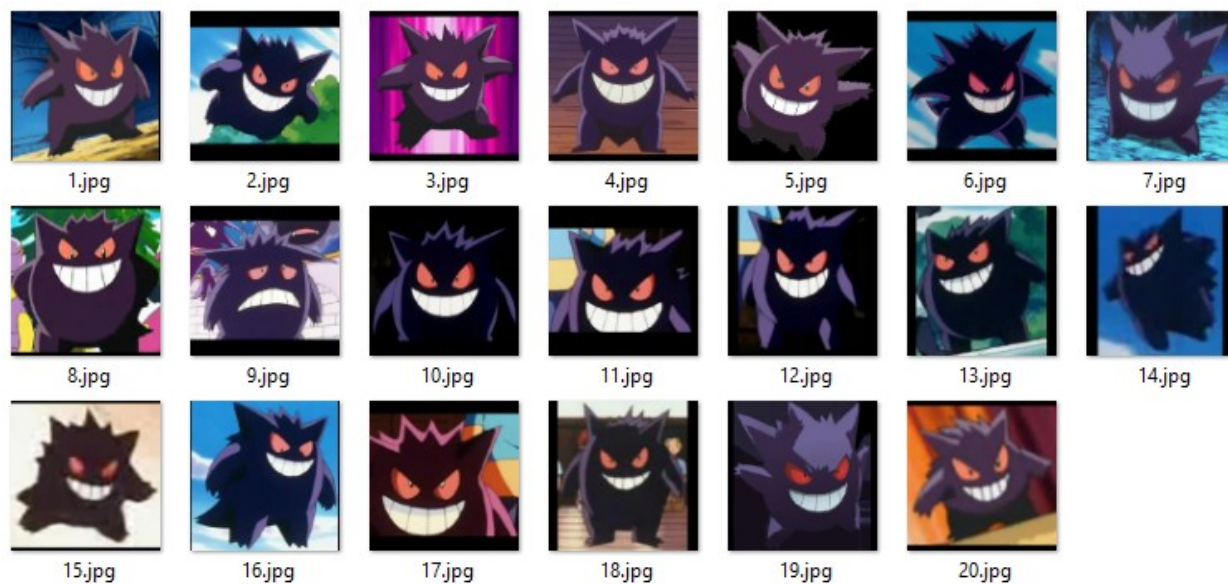
Приложения

Приложение 1. Примеры обучающих изображений

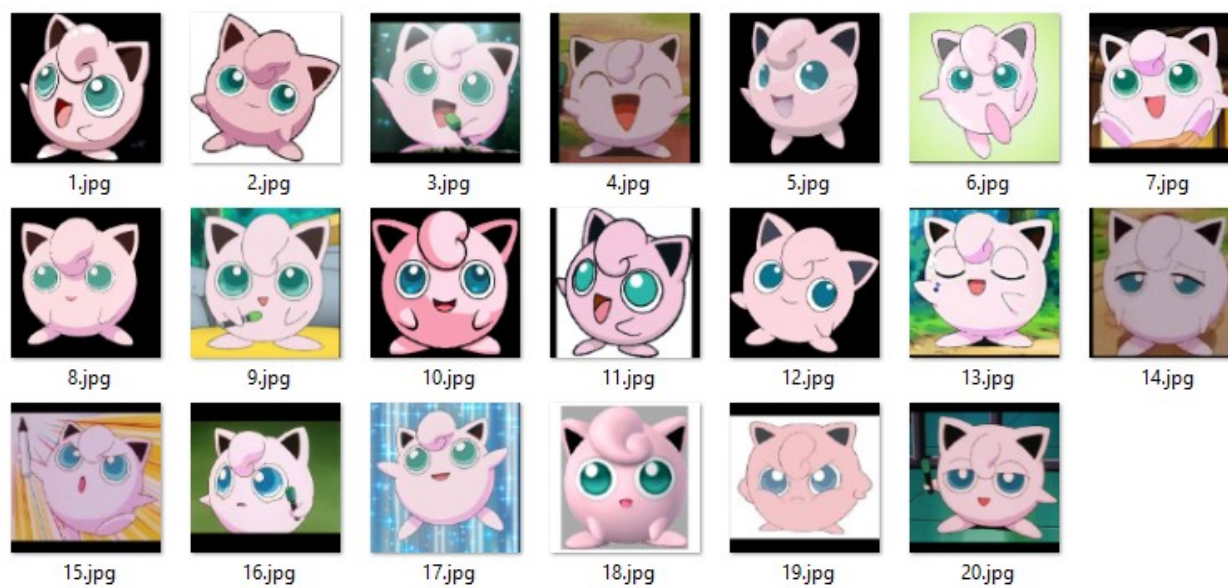
Бульбазавр



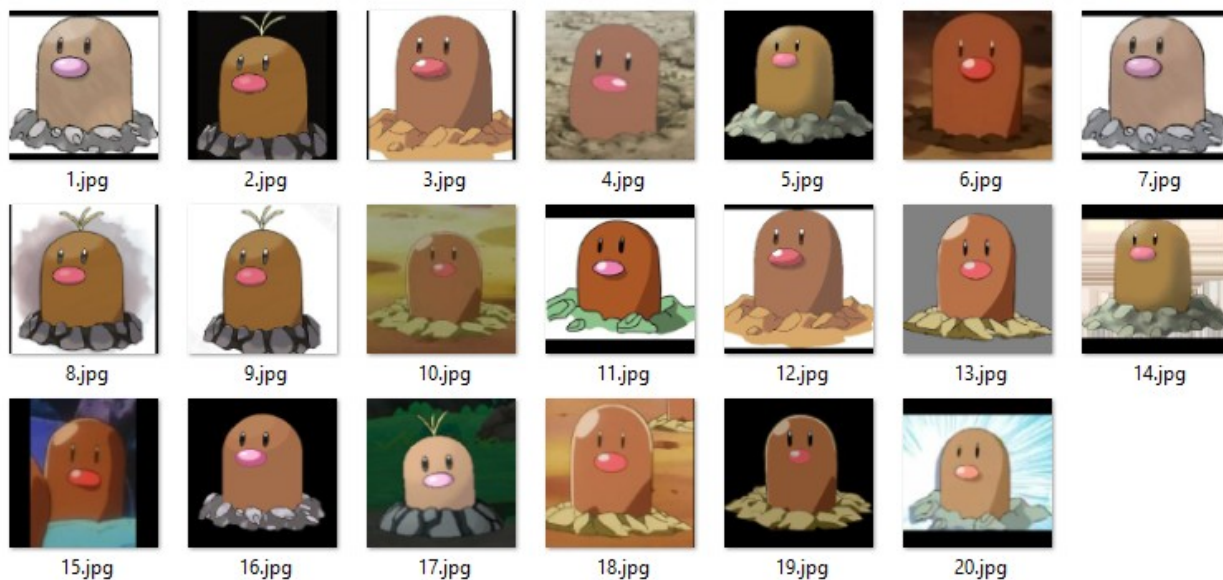
Генгар



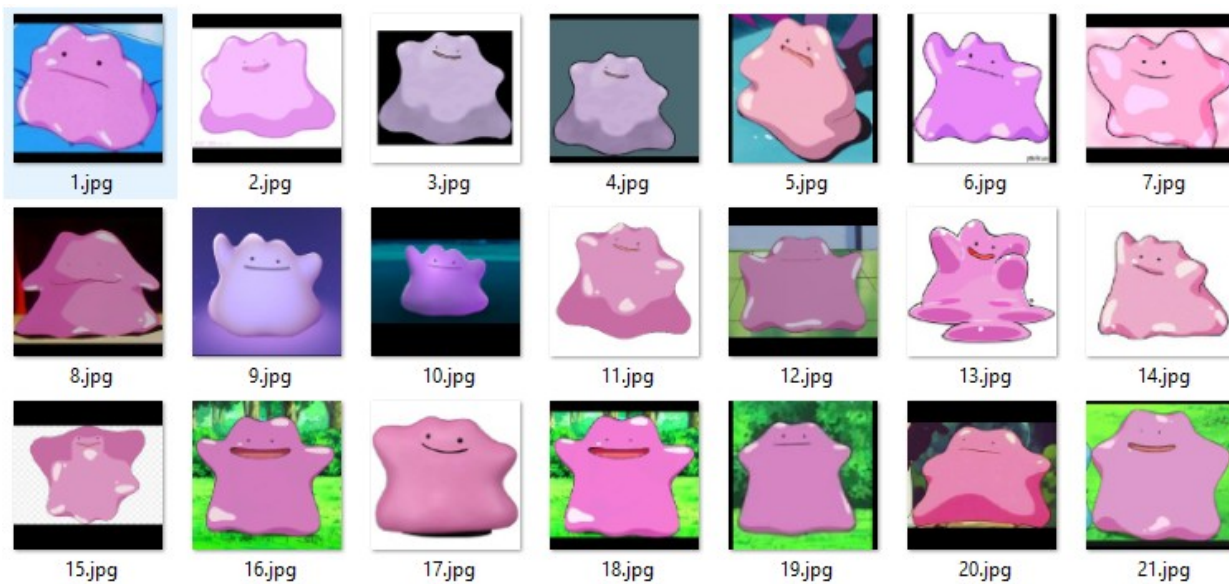
Джиггглипафф



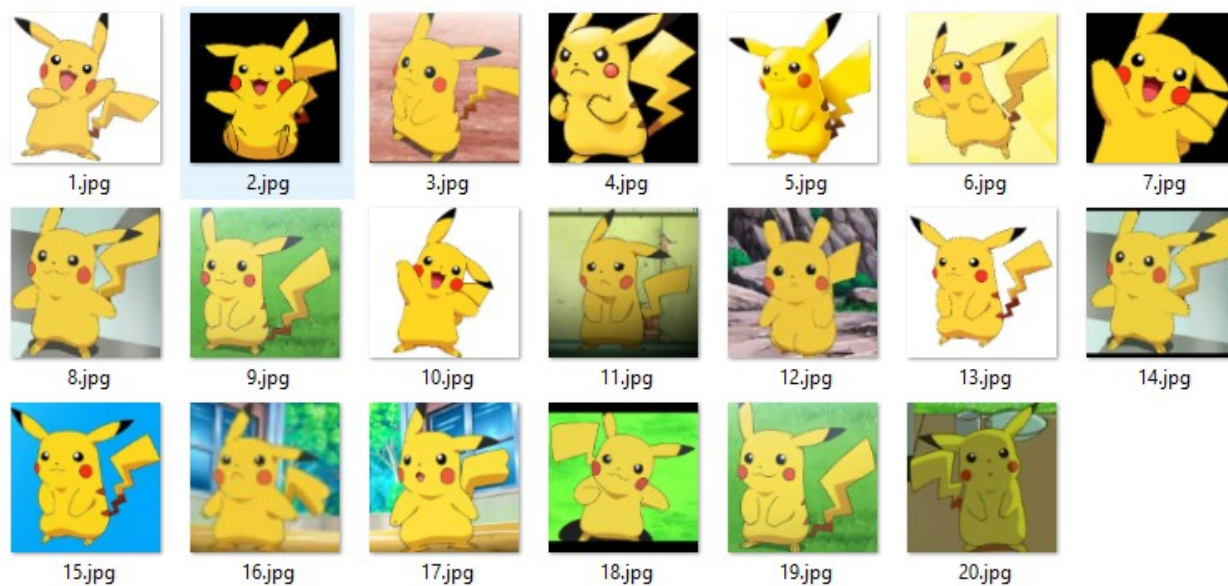
Дигглет



Дитто



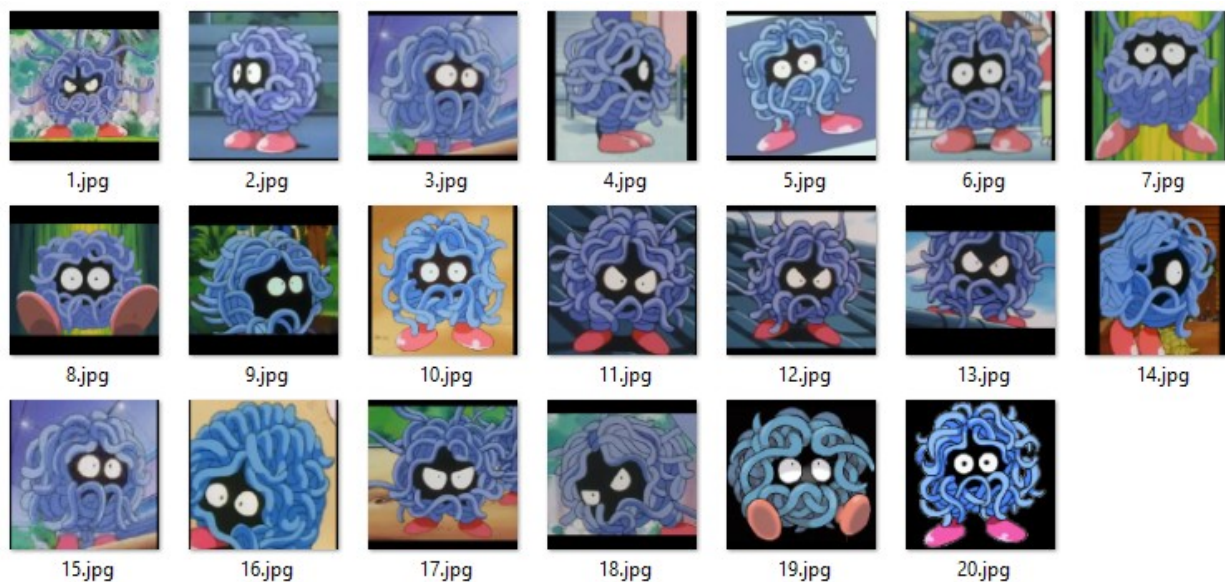
Пикачу



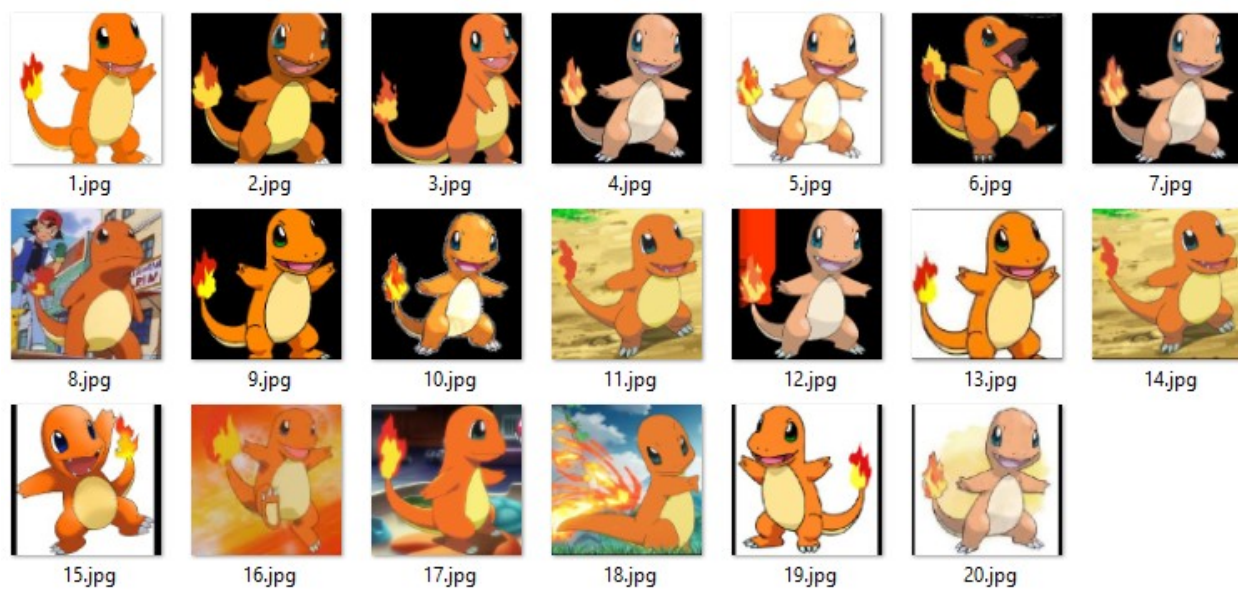
Сквиртл



Тангела



Чармандер



Черизард



Список литературы

<https://medium.com/@dimart/pok%C3%A9mon-recognition-d3ad5cad61e> - пример программы распознавания покемонов

<https://habr.com/ru/company/mlclass/blog/247751/>

<https://habr.com/ru/post/264241/>

https://scikit-learn.org/0.20/modules/model_persistence.html - сохранение модели на диск, для дальнейшего использования

https://scikit-learn.org/0.20/auto_examples/plot_anomaly_comparison.html#sphx-glr-auto-examples-plot-anomaly-comparison-py - сравнение некоторых методов классификации

<http://www.machinelearning.ru/wiki/images/6/6d/Voron-ML-1.pdf> - о машинном обучении

https://scikit-image.org/docs/dev/auto_examples/features_detection/plot_hog.html#sphx-glr-auto-examples-features-detection-plot-hog-py - описание и пример HOG-преобразования (определение мест и направления изменения яркости на изображении для выделения контуров)