Московский авиационный институт

(национальный исследовательский университет)

Факультет № 8 «Информационные технологии и прикладная математика»Кафедра 806 «Вычислительная математика и программирование»

РЕФЕРАТ

по дисциплине «Фундаментальная информатика»

1 семестр

на тему “Телеграмм-бот, который определяет животное по изображению”

Студент:	Мозговой Н.Е.
Группа:	М8О-109Б-22
Преподаватель:	Сысоев М.А.
Подпись:	
Оценка:	

Москва, 2022

СОДЕРЖАНИЕ

1. ПЛАН ВЫПОЛНЕНИЯ РАБОТЫ

1. Поиск библиотеки которая позволит в полной мере использовать возможности телеграмма
2. Поиск и изучение готовых нейросетей подходящих для нашей цели
3. Изучение библиотеки PIL, которая будет работать с изображением
4. Изучение библиотеки aiogram, на которой написан наш бот
5. Использование библиотеки Translate для перевода результатов работы нейросети
6. Тестирование бота в телеграмме

2. НАПИСАНИЕ ПРОГРАММЫ

1. Aiogram – библиотека, позволяющая реализовать полный функционал телеграмм бота

Благодаря данной библиотеке мы можем создать собственного телеграмм бота

@dp.message_handler(commands=['start']) - команда для создания кнопки старт, с которой бот начнет работать

@dp.message_handler(content_types=['photo']) – реагирует на все присланные изображения

@dp.message_handler(content_types=['text'])- реагирует на присланный текст

2. Библиотека os: позволяет нам работать с файлами, путями к файлам и так далее. Нам нужен лишь один ее метод – os.getcwd(), чтоб получить путь к директории, в которой мы сейчас работаем. Так мы будем сохранять изображение, чтобы потом закинуть его в нейронку, а затем удалить.

Библиотека json: нужна чтоб открыть json файл, в котором хранятся результаты работы нейросети. Дело в том, что за неумением создавать полноценные новые нейросети, я взял модель типа CIFAR.Она уже обучена и хранится в библиотеке torchvision, откуда мы ее просто импортируем.

Библиотека translate нам нужна, потому что результат работы нейронка возвращает нам на английском, а мы английский плохо знаем, поэтому пришлось использовать костыли.

Ну и библиотека PIL нужна, чтоб загруженное изображение открыть в программе

Структура работы бота:

- пользователь запускает бота командой /start
- бот запрашивает у пользователя фото
- пользователь отправляет изображение(если он отправляет текстовое сообщение, бот ему об этом сообщает)
- после получение изображения, программа сохраняет его в папку с нейросетью. Изображение загружается в программу, потом его подготавливают к использовании в нашей модели. Далее изображение загружается в нейронку и она возвращает нам результат. Возвращает она его нам на английском, поэтому мы переводим результат на русский и бот отправляет пользователю этот самый результат

Код:

```
def predict(image):
    model = models.resnet18(pretrained=True)
    model.eval()

    out = model(image)


    _, pred = torch.max(out, 1)
    idx_to_label = get_idx_to_label()
    cls = idx_to_label[str(int(pred))]
    translation = translator.translate(cls)
    return translation
```

-написана функция predict для получения результата работы нейросети

```
def load_image():
    image = Image.open('test.jpg')
    transform = get_image_transform()
    image = transform(image)[None]
    return image
```

-функция load_image вызывает внутри себя функцию get_image_transform, которая трансформирует отправленное изображение относительно параметров нейросети

```
def get_idx_to_label():
    with open("imagenet_idx_to_label.json") as f:
        return json.load(f)
```

-функция которая открывает файл с данными

```
if __name__ == '__main__':
    executor.start_polling(dp)
```

- последняя команда для того, чтобы бот работал, при запуске программы.

ЗАКЛЮЧЕНИЕ

Благодаря проделанной работе, я получил большой опыт в работе с тг-ботами, стал лучше работать и использовать библиотеки питона, а также лучше понимать сам язык.

Работа очень понравилась, безусловно, приобретенные навыки в ходе этой работы могут понадобиться в ближайшем будущем.