Приложение на Django для распознавания объектов

Содержание

[Обзор проекта 1](#_heading=h.gjdgxs)

[Структура проекта 2](#_heading=h.30j0zll)

[Домашняя страница 2](#_heading=h.1fob9te)

[Дашборд 2](#_heading=h.3znysh7)

[Страница авторизации 4](#_heading=h.2et92p0)

[Шаблоны страниц 5](#_heading=h.tyjcwt)

[Заключение 6](#_heading=h.3dy6vkm)

[Приложение 1. Пример файловой структуры проекта 7](#_heading=h.1t3h5sf)

[Приложение 2. Список необходимых библиотек 8](#_heading=h.4d34og8)

# Обзор проекта

Предлагается сделать веб приложение на базе фреймворка Django, которое позволит пользователям применять предобученную модель для обнаружения и классификации объектов на фото.

Соответственно, потребуются механизмы регистрации и авторизации пользователей, чтобы пользователи могли создавать аккаунты и видеть только собственноручно загруженные изображения. Для интерфейса используется Bootstrap.

К использованию предлагается модель [MobileNet SSD](https://github.com/chuanqi305/MobileNet-SSD). Она определяет следующие классы:

* самолёт
* велосипед
* птица
* лодка
* бутылка
* автобус
* автомобиль
* кот
* стул
* корова
* стол
* собака
* лошадь
* мотоцикл
* человек
* цветок
* овца
* диван
* поезд
* телевизор

В оригинале - "aeroplane", "bicycle", "bird", "boat", "bottle", "bus", "car", "cat", "chair", "cow", "diningtable", "dog", "horse", "motorbike", "person", "pottedplant", "sheep", "sofa", "train", "tvmonitor" (также есть отдельный класс для фона).

# Структура проекта

Проект включает следующие ключевые компоненты:

## Домашняя страница

Тут есть ссылка на страницу с обработанными фотографиями (далее – дашборд). Туда есть доступ только у авторизованноых пользователей, так что при попытке зайти в дашборд до захода в аккаунт, должно совершаться перенаправление на страницу авторизации. С неё можно зайти в аккаунт или создать новый.

A screenshot of a computer

Description automatically generated

*Рисунок 1 Домашняя страница*

## Дашборд

Тут можно загружать фотографии, обрабатывать их и ознакомиться с результатами уже законченной обработки. Результат представляет из себя название класса обнаруженного объекта и уверенность в полученном результате. Сам же обнаруженный объект обводится рамкой. Также можно удалить фото.

A screenshot of a computer

Description automatically generated

*Рисунок 2 Пример экрана обработки изображений. Вверху - уже обработанное, с подписью о результате обработки.*

A screenshot of a computer

Description automatically generated

*Рисунок 3 При нажатии на миниатюру фото оно открывается в новой вкладке, в полный размер*

A screenshot of a computer

Description automatically generated

*Рисунок 4 Пример с уверенностью меньшей чем 100% (0.98 = 98%)*

A screenshot of a computer

Description automatically generated

*Рисунок 5 После удаления обработанных фото*

## Страница авторизации

Здесь пользователи могут зайти в учётную запись. Если её нет – создать (Register – перенаправляет на форму регистрации).

A screenshot of a computer

Description automatically generated

*Рисунок 6 Экран входа в учётную запись*

A screenshot of a computer

Description automatically generated

*Рисунок 7 Экран регистрации*

## Шаблоны страниц

base.html – базовый шаблон, подключающий Bootstrap, на его основании пишутся остальные. Отдельно стоит отметить что в нём прописана навигационная панель

home.html – домашняя страница, со ссылкой на дашборд

add\_image\_feed – загрузка пользователем изображения

dashboard.html – тут можно перейти на экран загрузки, а также для всех уже загруженных кртинок текущим пользователем, посмотреть фото и обработать. Либо если уже обработаны, ознакомиться с результатами обработки.

login.html – вход в учётную запись

registration.html – создание учётной записи

logout.html – выход из учётной записи

# Заключение

Это приложение предоставляет полнофункциональную платформу для пользователей, которым необходимо выполнять задачи по обнаружению объектов на изображениях. Благодаря использованию готовых моделей глубокого обучения и удобного веб-интерфейса, проект отлично подходит для пользователей без опыта работы с машинным обучением. Одним из вариантом расширения функционала может быть подключение нескольких моделей и предоставление выбора, какую модель использовать.

Например, много моделей можно найти на [huggingface](https://huggingface.co/)

# Приложение 1. Пример файловой структуры проекта

│ db.sqlite3

│ manage.py

│

├───detection\_site

│ │ asgi.py

│ │ settings.py

│ │ urls.py

│ │ wsgi.py

│ │ \_\_init\_\_.py

│ │

│

├───media

│ ├───images

│ │

│ └───processed\_images

│

└───object\_detection

│ admin.py

│ apps.py

│ forms.py

│ mobilenet\_iter\_73000.caffemodel

│ mobilenet\_ssd\_deploy.prototxt

│ models.py

│ tests.py

│ urls.py

│ utils.py

│ views.py

│ \_\_init\_\_.py

│

├───migrations

│ │ 0001\_initial.py

│ │ 0002\_imagefeed\_processed\_image.py

│ │ \_\_init\_\_.py

│

├───templates

│ └───object\_detection

│ add\_image\_feed.html

│ base\_generic.html

│ dashboard.html

│ home.html

│ login.html

│ register.html

# Приложение 2. Список необходимых библиотек

Django==5.0.4

numpy==1.26.4

opencv-python-headless==4.9.0.80

pillow==10.3.0

sqlparse==0.5.0

typing-extensions==4.11.0

tzdata==2024.1