Приложение на Django для распознавания объектов

Содержание

[Обзор проекта 1](#_Toc175604512)

[Структура проекта 2](#_Toc175604513)

[Домашняя страница 2](#_Toc175604514)

[Дашборд 2](#_Toc175604515)

[Страница авторизации 2](#_Toc175604516)

[Шаблоны страниц 2](#_Toc175604517)

[Заключение 3](#_Toc175604518)

[Приложение 1. Пример файловой структуры проекта 3](#_Toc175604519)

[Приложение 2. Список необходимых библиотек 4](#_Toc175604520)

Обзор проекта

Предлагается сделать (доработать) веб приложение на базе фреймворка Django, которое позволит пользователям применять предобученную модель для обнаружения и классификации объектов на фото и видео с камеры.

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 для распознавания изображений [mobilenet\_iter\_73000.caffemodel](https://programmersought.com/article/65006829867/). 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).

А так же модель для распознания лиц работающая в видеоряде со встроенной камеры ноутбука [haarcascade\_profileface.xml](https://arrrsite.wordpress.com/2018/06/13/face-recognition-with-opencv-and-python/). Она определяет:

* лица

Структура проекта

Проект включает следующие ключевые компоненты:

Домашняя страница

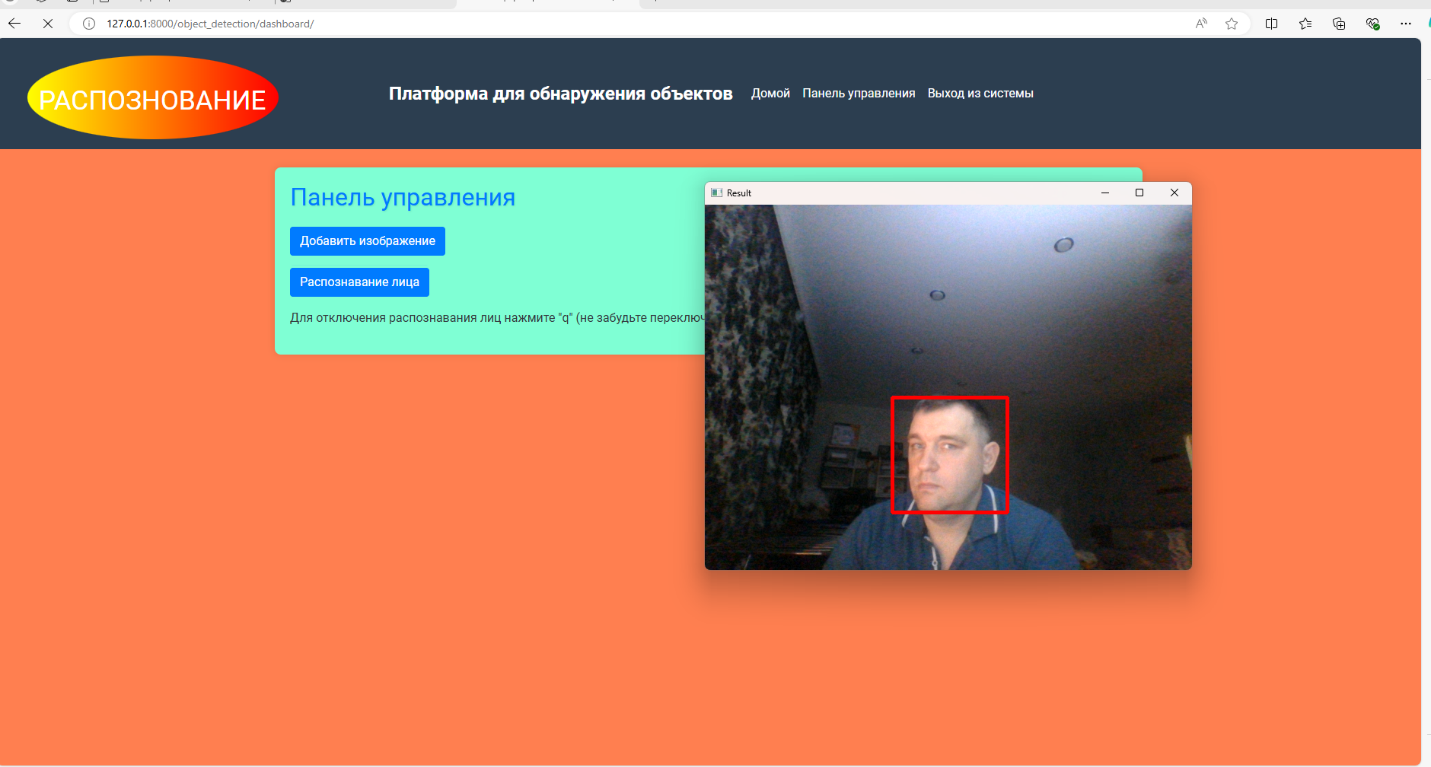
Тут есть ссылка на страницу с обработанными фотографиями (далее – дашборд). Туда есть доступ только у авторизованноых пользователей, так что при попытке зайти в дашборд до захода в аккаунт, должно совершаться перенаправление на страницу авторизации. С неё можно зайти в аккаунт или создать новый.

*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. Можно удалить фото.

*Рисунок 2. Домашняя страница. Дашборд.*

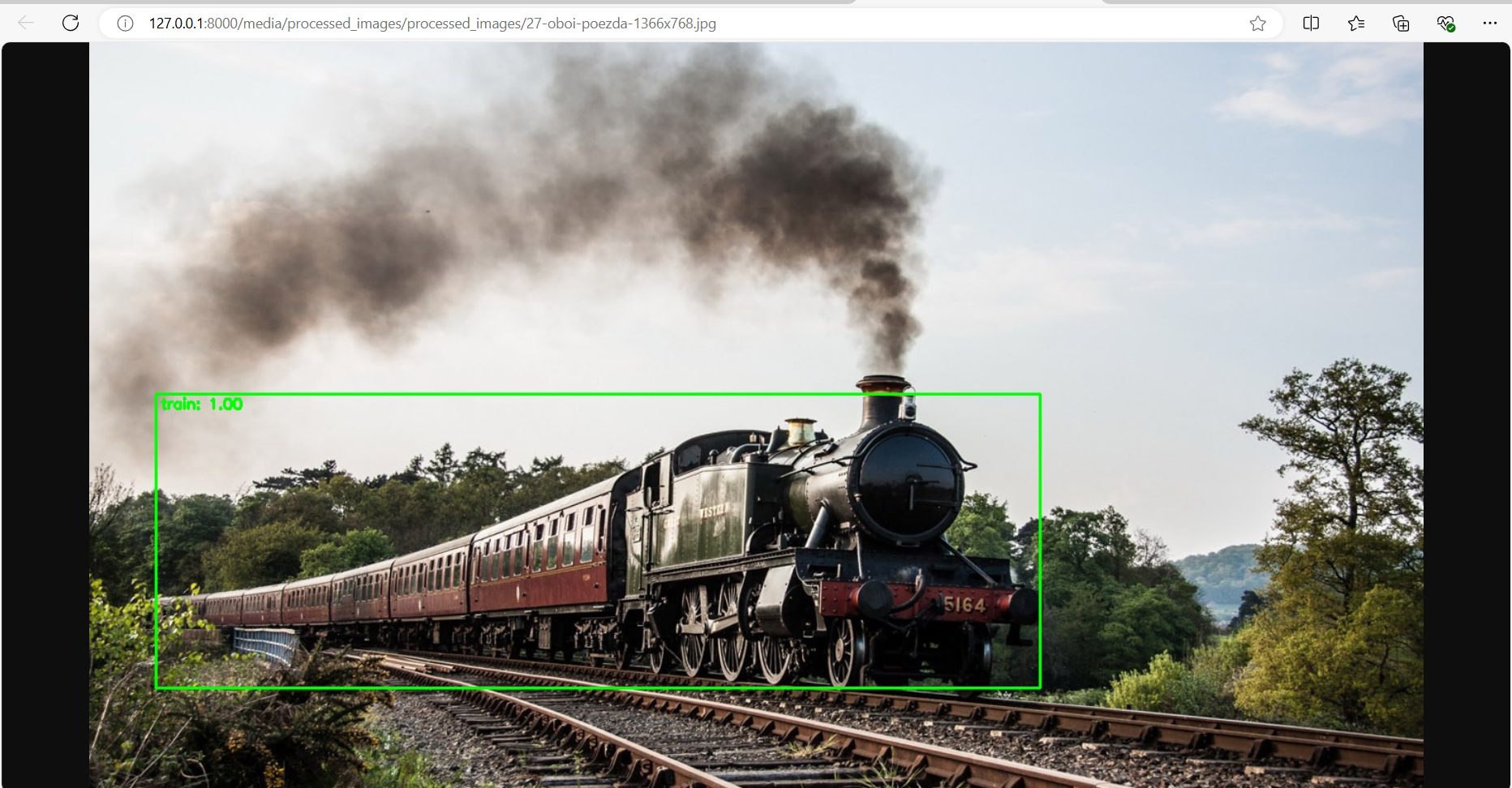


А также запускать видео со встроенной камеры в видеоряде которой работает модель распознавания лиц. При этом распознанные лица обводятся красным прямоугольником.

*Рисунок 3. Экран обработки изображений с подписью о результате обработки.*



*Рисунок 4. При нажатии на миниатюру фото оно открывается в новой вкладке, в полный размер.*

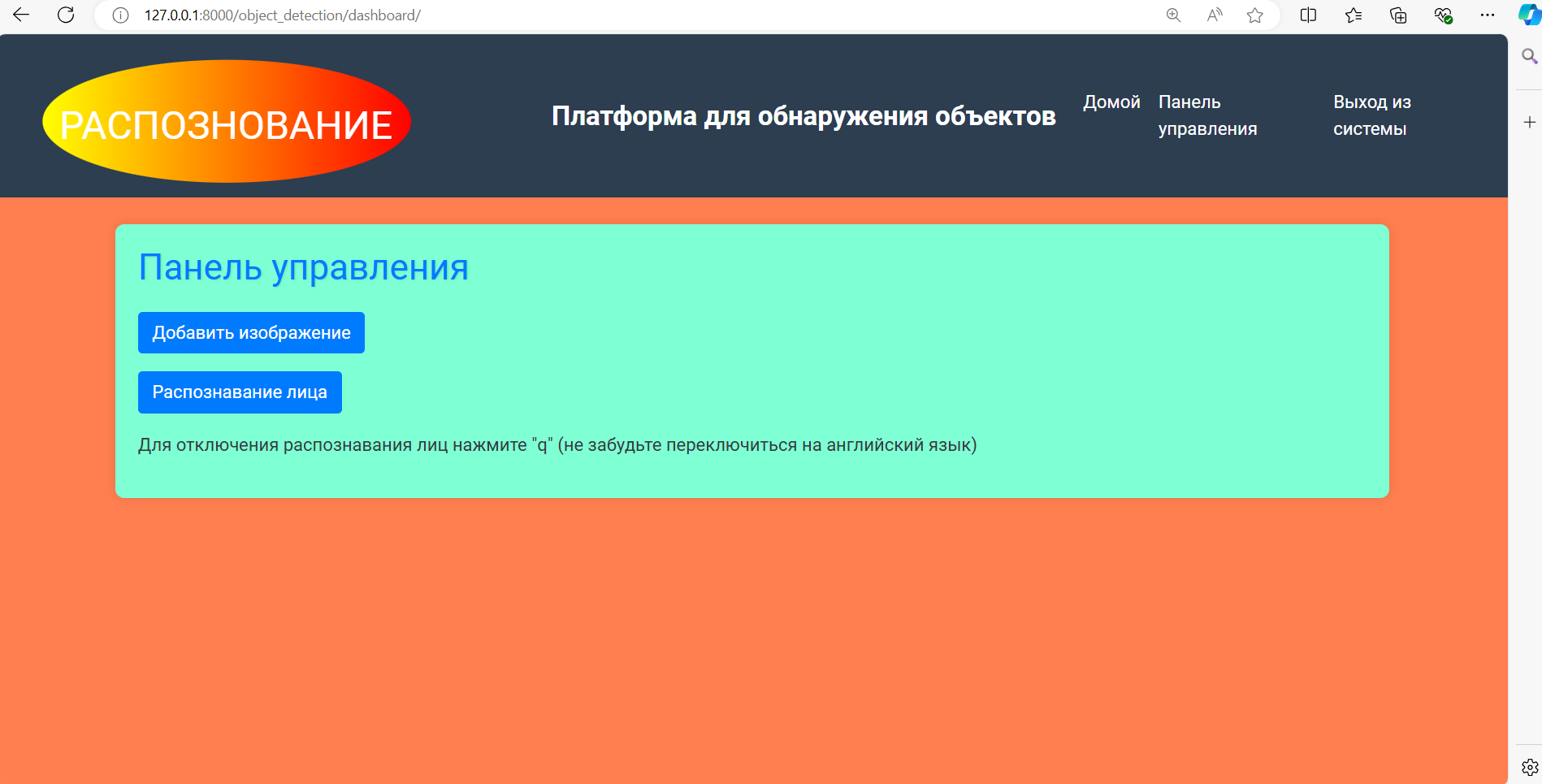


*Рисунок 6. Пример с уверенностью меньшей чем 100% (0.98 = 98%).*



В примере видно, что вероятность распознания менее 100%. Котопес определен как стул.

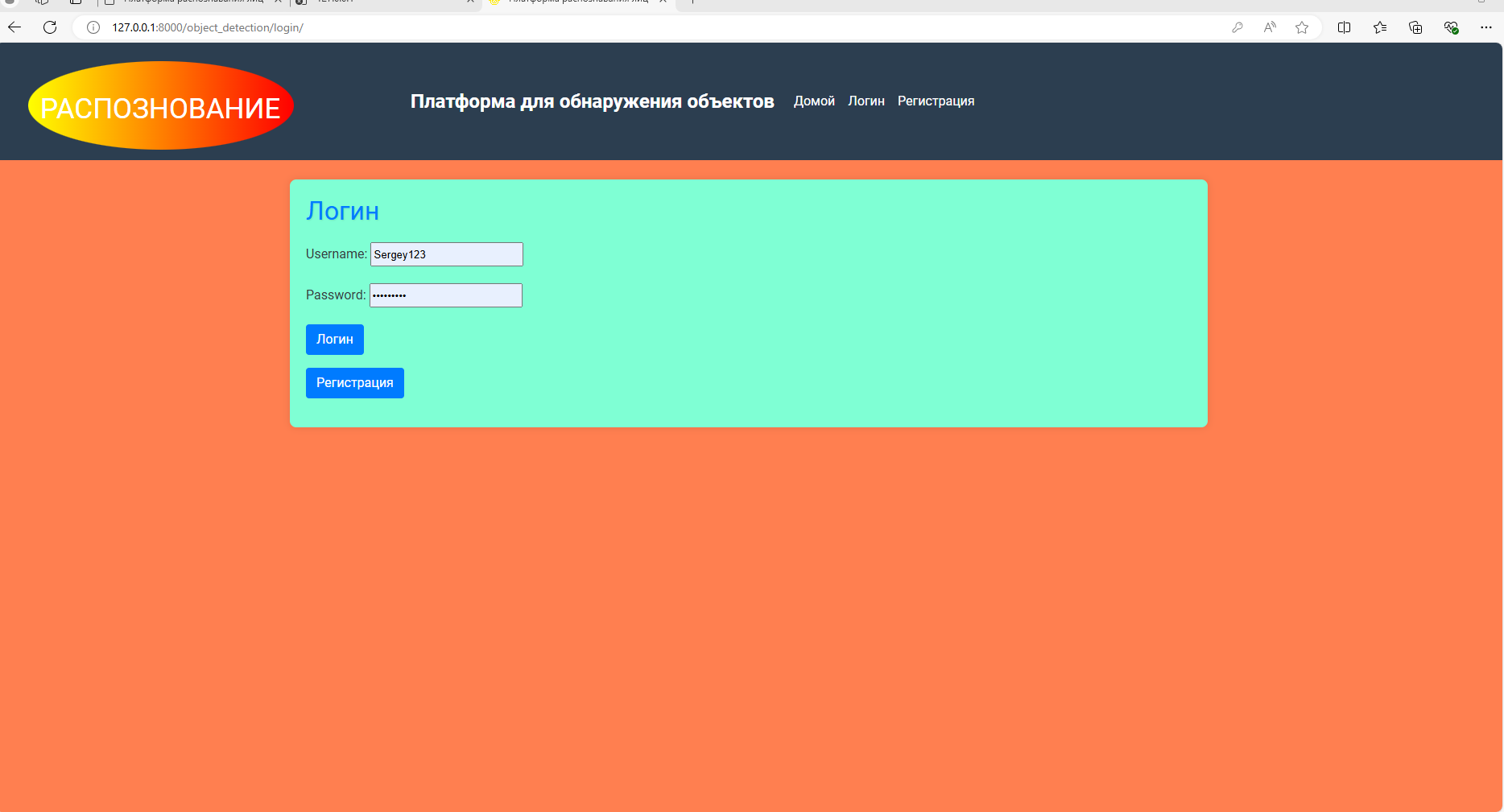
*Рисунок 7. После удаления обработанных фото.*



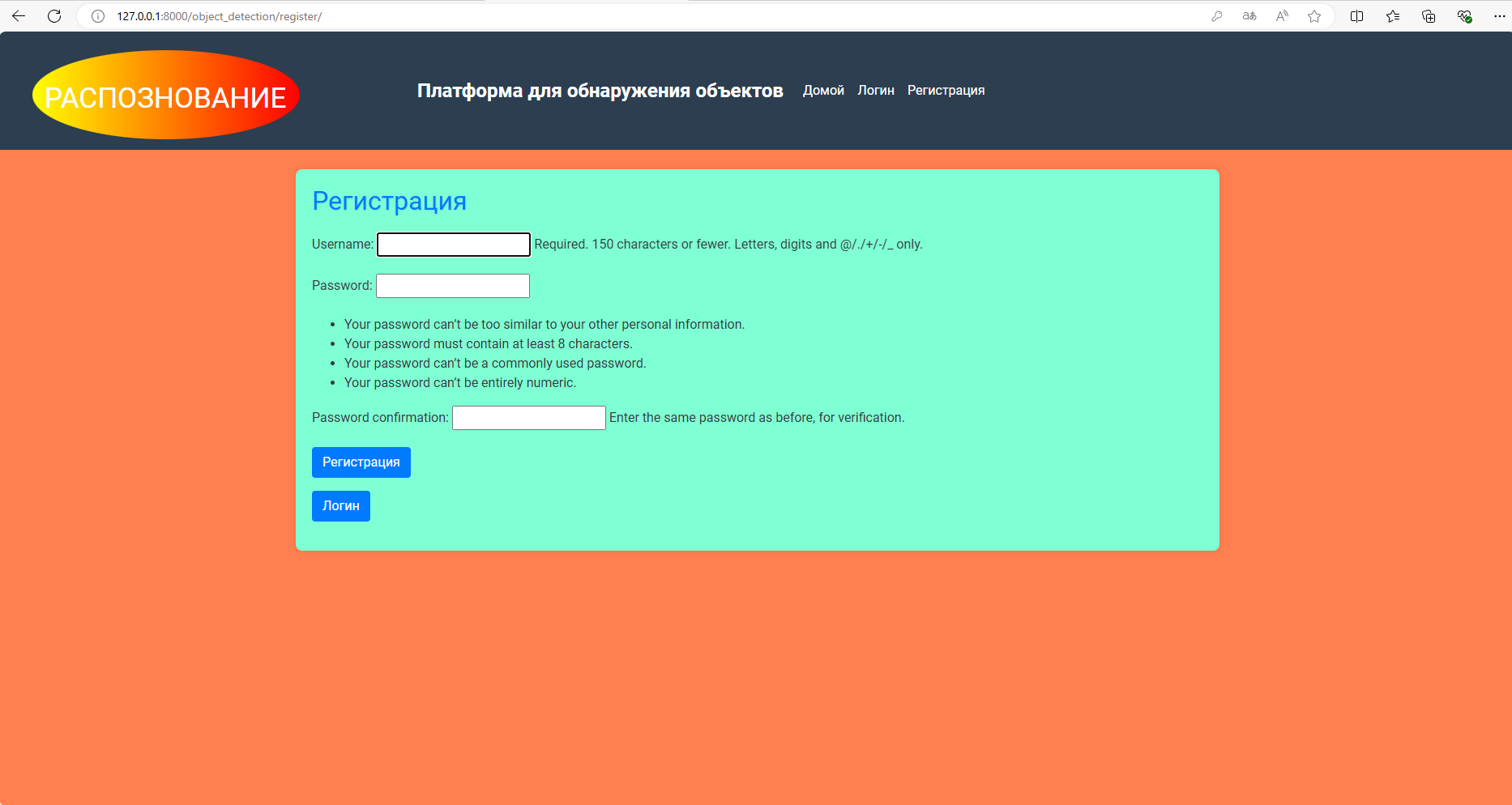
Страница авторизации

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

*Рисунок 8. Экран входа в учётную запись.*



*Рисунок 9. 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

│ haarcascade\_profileface.xml

    │   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

asgiref==3.8.1

channels==4.1.0

Django==5.0.4

numpy==1.26.4

opencv-python==4.8.1.78

pillow==10.3.0

sqlparse==0.5.0

typing\_extensions==4.11.0

tzdata==2024.1