«Urban University»

Дипломный проект

**Практическое задание по теме:**

Приложение на Django для распознавания объектов используя две предобученые модели.

Исполнитель: Кандинский К.А.

Томск

2024

Приложение на Django для распознавания объектов используя две предобученые модели.

Содержание

Обзор

Структура проекта

Дашборд

Страница регистрации

Страница авторизации

Шаблоны страниц

Заключение

Приложение 1 Файловая структура проекта

Приложение 2 Список необходимых библиотек

Приложение 3 Список использованных источников

**Обзор**

В данном проекте планируется сделать приложение с использованием фремворка Django для распознавания и классификации объектов на фотографии используя две предобученые модели: MobileNet SSD и YOLO.

Для разграничения прав доступа потребуется механизмы регистрации и авторизации пользователей, чтобы пользователи могли видеть только свои загруженные изображения.

Для распознания и классификации предлагается две модели, MobileNet SDD и YOLO.

У каждой модели разное количество классов для распознования.

Соответственно у MobileNET SSD:

class\_names = [ "background", "aeroplane", "bicycle", "bird", "boat","bottle", "bus", "car", "cat", "chair","cow", "diningtable", "dog", "horse", "motorbike", "person", "pottedplant", "sheep", "sofa", "train","tvmonitor"]

У YOLO :

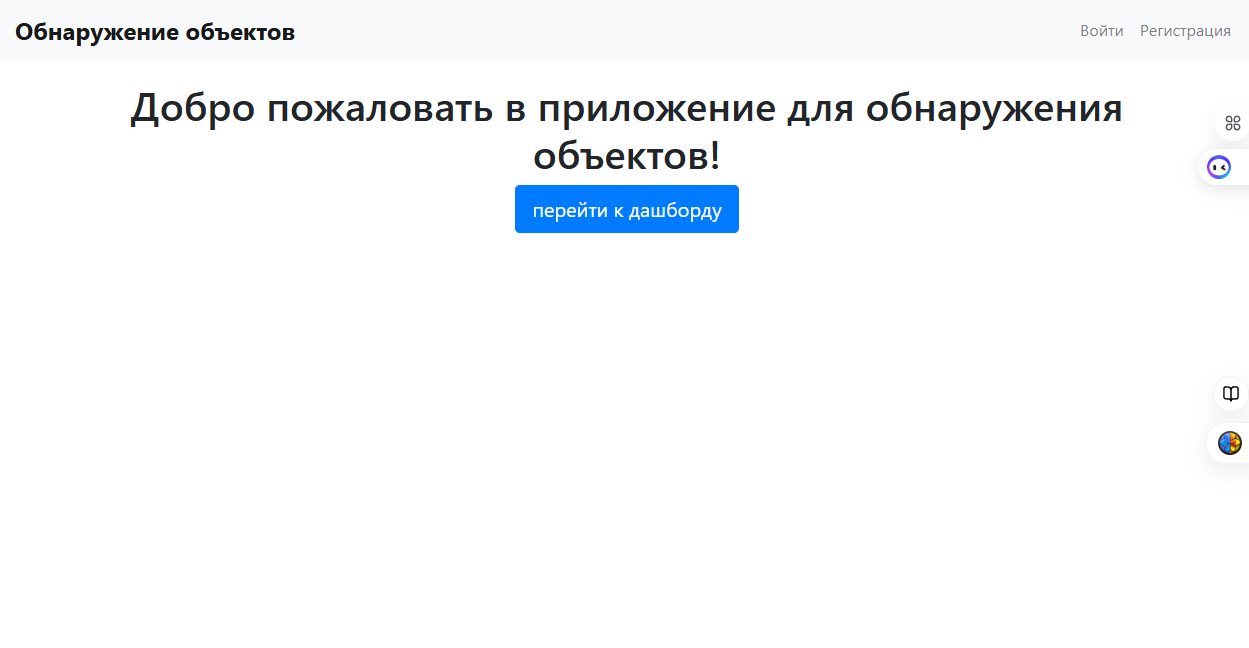
class\_names = [‘person’, ‘bicycle’, ‘car’, ‘motorbike’, ‘aeroplane’, ‘bus’, ‘train’, ‘truck', ‘boat’, ‘traffic light’, ‘fire hydrant’, ‘stop sign’, ‘parking meter’, ‘bench’, ‘bird’, ‘cat', ‘dog’, ‘horse’, ‘sheep’, ‘cow’, ‘elephant’, ‘bear’, ‘zebra’, ‘giraffe’, ‘backpack’, ‘umbrella’, ‘handbag’, ‘tie’, ‘suitcase’, ‘frisbee’, ‘skis’, ‘snowboard’, ‘sports ball’, ‘kite’, ‘baseball bat’, ‘baseball glove’, ‘skateboard’, ‘surfboard’, ‘tennis racket’, ‘bottle’, ‘wine glass’, ‘cup’, ‘fork’, ‘knife’, ‘spoon’, ‘bowl’, ‘banana’, ‘apple’, ‘sandwich’, ‘orange’, ‘broccoli’, ‘carrot’, ‘hot dog’, ‘pizza’, ‘donut’, ‘cake’, ‘chair’, ‘sofa’, ‘pottedplant’, ‘bed’, ‘diningtable’, ‘toilet’, ‘tvmonitor’, ‘laptop’, ‘mouse’, ‘remote’, ‘keyboard’, ‘cell phone’, ‘microwave’, ‘oven’, ‘toaster’, ‘sink’, ‘refrigerator’, ‘book’, ‘clock’, ‘vase’, ‘scissors’,‘teddy bear’,‘hair drier’,‘toothbrush’]

Как видим YOLO распознает в несколько раз больше классов чем MobileNet. Но и занимает в 10 раз больше что должно сказаться на скорости работы.

**Структура проекта**

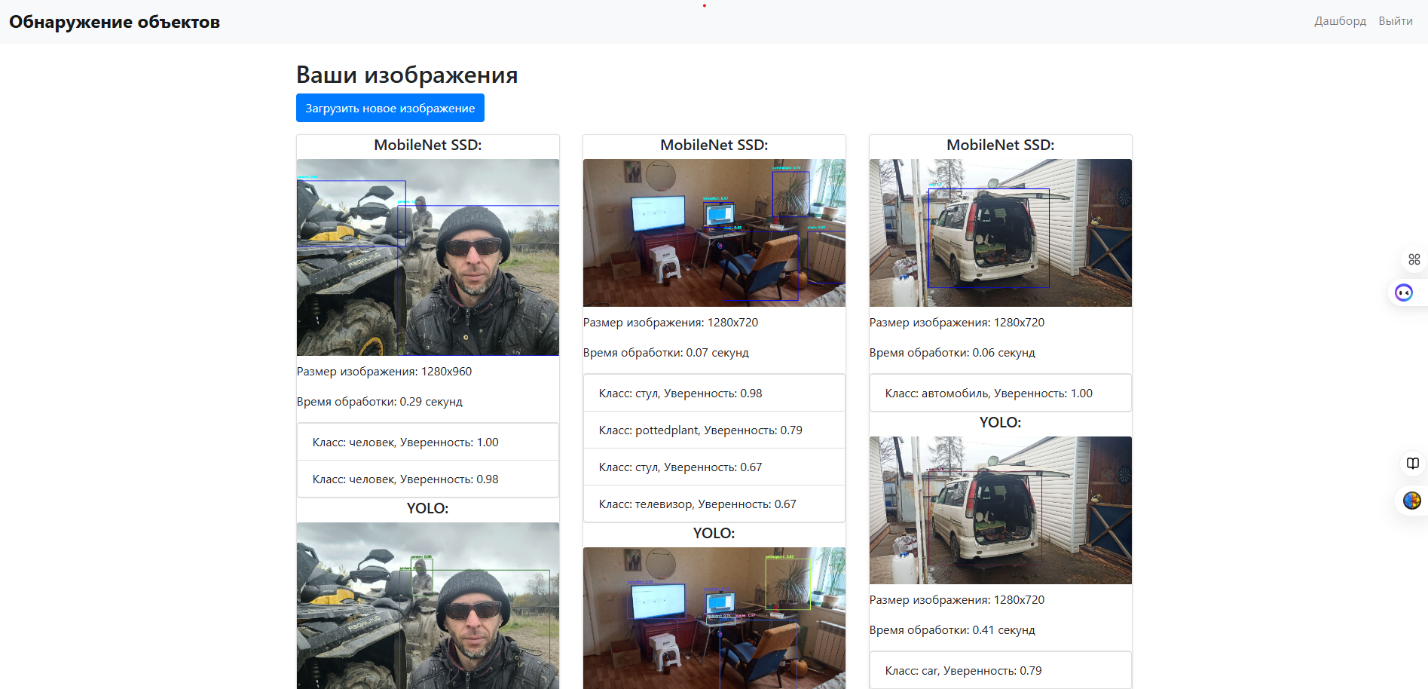
*Домашняя страница*

На данной странице имеются ссылки на вход в аккаунт для зарегистрированных пользователей и ссылка на страницу регистрации.



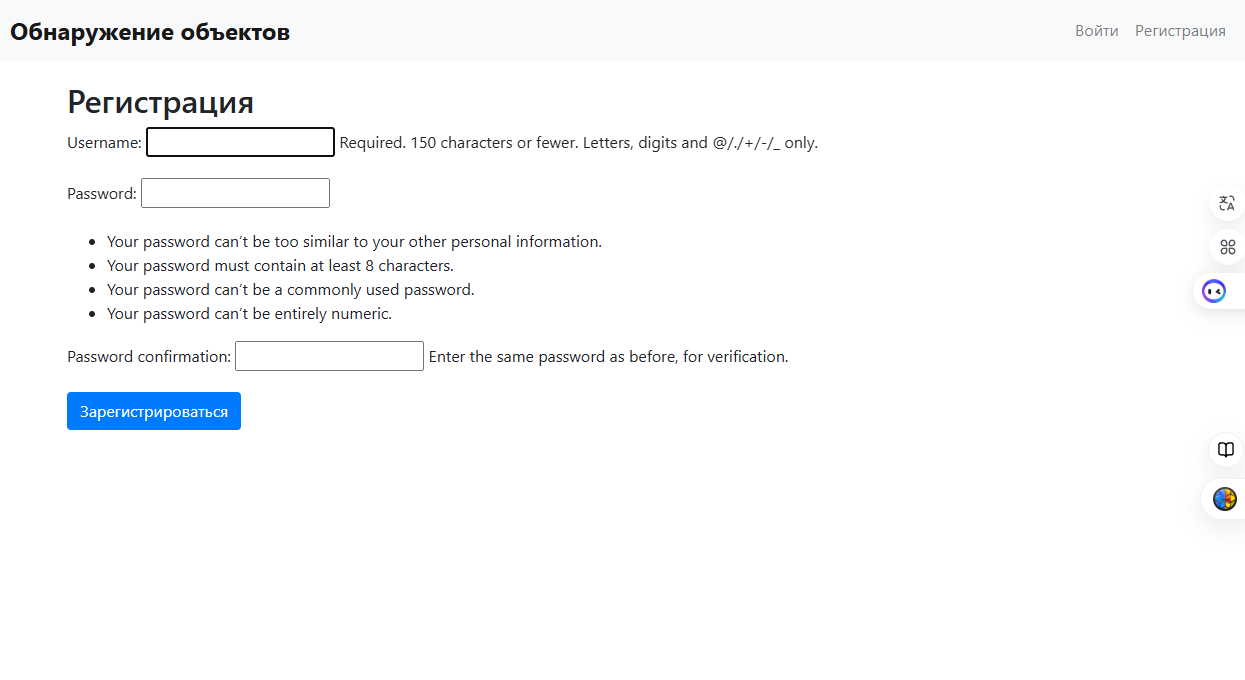
*Дашборд*

Основная страница для обработки, доступна только для зарегистрированных пользователей. Здесь можно загружать фотографии, запускать процесс обработки и удалять загруженные фотографии. После обработки под фотографией для каждой модели отображается размер изображения, время обработки а также найденные и классифицированные объекты с точностью обнаружения для каждого объекта.



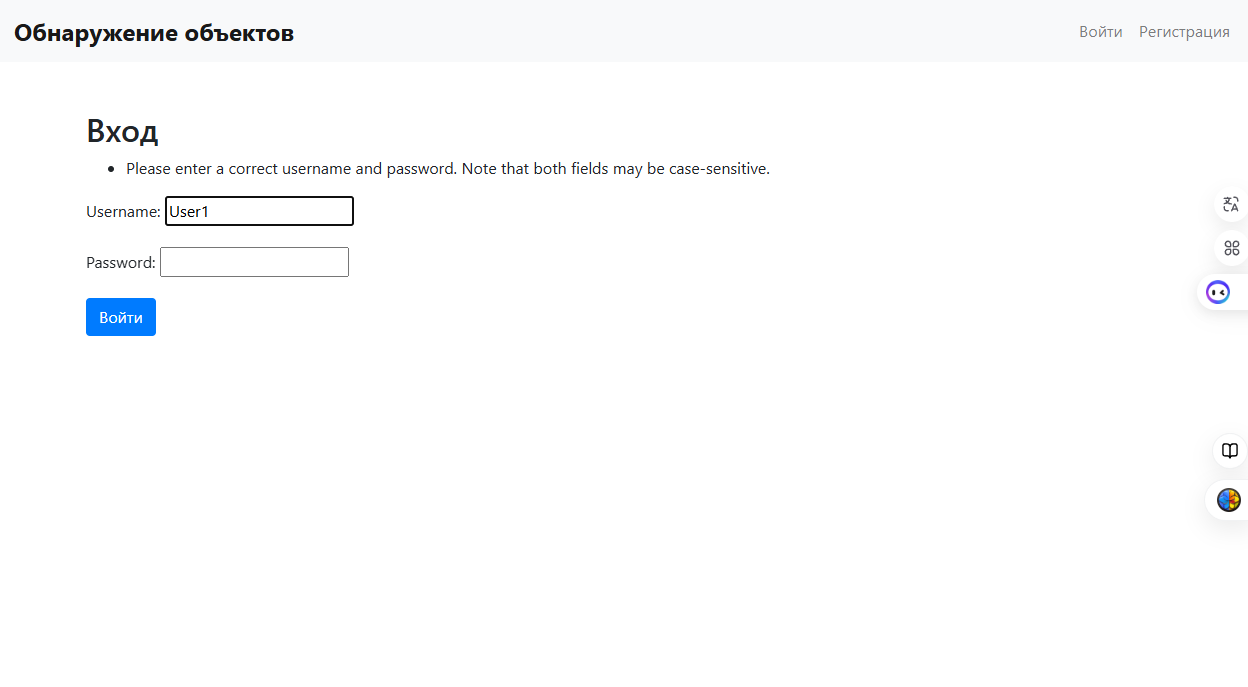
*Страница регистрации*

На данной странице новый пользователи могут создать учетную запись. Также здесь проверяется корректность введенных данных.



*Страница авторизации*

На данной странице зарегистрированные пользователи могут войти в свою учетную запись. Если введены не корректные данные или пользователе не существует, то происходит перенаправление на предыдущую страницу.



*Шаблоны страниц*

add\_image\_feed.html – шаблон для добавления изображений.

base.html – базовый шаблон в котором подключается bootstrap, все остальные страницы пишутся на его основе.

dashboard.html – в данном шаблоне можно загрузить изображения, обработать если еще не обработаны, посмотреть результаты обработанных изображений или удалить изображения.

home.html – домашняя страница со ссылкой на дашборд.

login.html – страница авторизации.

logout.html – выход из учетной записи.

registration.html – шаблон для регистрации новых пользователей.

**Заключение**

В заключении хотелось бы сравнить две модели и дать рекомендации по их применению. А также сделать обзор и дать рекомендации по применению фреймворка Django.

1. MobileNet SSD

Описание:

- MobileNet SSD (Single Shot MultiBox Detector) - это легковесная модель, оптимизированная для мобильных и встроенных систем. Она сочетает в себе архитектуру MobileNet, которая известна своей эффективностью для задачи обнаружения объектов.

Преимущества:

- Высокая скорость: Ориентирована на реальное время, что делает её подходящей для приложений, где задержка важна.

- Низкие требования к ресурсам: Благодаря упрощённой архитектуре, она идеально подходит для устройств с ограниченными вычислительными возможностями.

Недостатки:

- Точность: Уступает более сложным моделям, как, например, YOLO или более новые версии моделей семьи Detectors.

2. YOLO (You Only Look Once)

Описание:

- YOLO - это одноступенчатая модель, которая обрабатывает изображение целиком и одновременно предсказывает границы и классы объектов. Это делает её чрезвычайно быстрой и эффективной.

Преимущества:

- Совершенствование в точности: Показала себя более точно в различных задачах, чем другие быстрые модели обнаружения.

- Быстрая: Благодаря своему подходу, она обрабатывает изображения быстрее, чем многие другие системы.

Недостатки:

- Ресурсоёмкость: Более ресурсоёмкая по сравнению с MobileNet SSD, требует более мощных устройств для максимальной производительности.

Рекомендации по применению

1. MobileNet SSD может быть рекомендована для:

- Мобильных приложений: Где важна скорость и эффективность.

- Встроенных систем: Где оборудование имеет ограничения по памяти и мощности.

- Приложений в области безопасности: Например, в системах умного дома, где важно быстрое обнаружение.

2. YOLO может быть рекомендована для:

- Дронов и робототехнических систем: Где важна высокая точность в реальном времени.

- Сложных задач компьютерного зрения: Например, для детекции в автономных транспортных средствах.

- Приложений в области медицинской диагностики: Где требуется высокая точность обнаружения патологий.

Описание фреймворка Django

Django — это высокоуровневый веб-фреймворк на языке программирования Python, который позволяет быстро и просто создавать безопасные и поддерживаемые веб-приложения. Основная философия Django заключается в содействии удобству разработки и увеличении производительности. Он предоставляет широкий набор инструментов и библиотек для выполнения самых различных задач в веб-разработке.

Основные особенности Django:

- Шаблоны: Django использует мощный шаблонизатор, который позволяет легко отделить логику обработки данных от представления.

- ORM (Object-Relational Mapping): Позволяет работать с базами данных, используя Python-классы вместо работы с сырым SQL.

- Административная панель: Генерация удобной админ-панели автоматически на основе моделей данных.

- Аутентификация и авторизация: Встроенные механизмы для управления пользователями и их доступами.

- Поддержка формы и валидации: Удобные инструменты для обработки пользовательских ввода.

Рекомендации по применению Django

Django наиболее эффективен в следующих областях и сценариях:

1. Корпоративные приложения: Благодаря архитектурным и административным возможностям, Django хорошо подходит для создания сложных корпоративных систем с многоуровневым управлением доступом.

2. Контент-ориентированные сайты: Сайты, которые во многом зависят от представления и управления контентом, такие как блоги или новостные порталы.

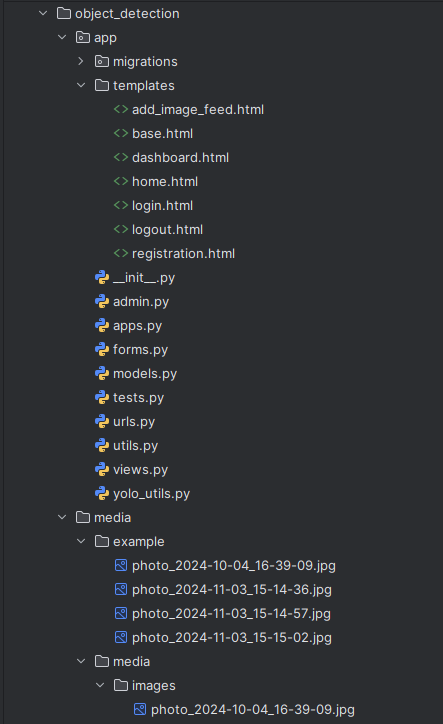
3. Прототипирование: Быстрая разработка и тестирование прототипов из-за своей простоты и лаконичности.

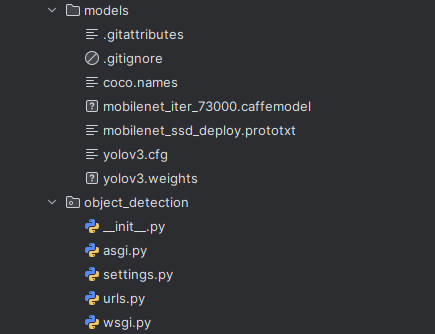
4. Проекты с сложной бизнес-логикой: Возможность легко моделировать бизнес-логику благодаря богатым возможностям ORM.

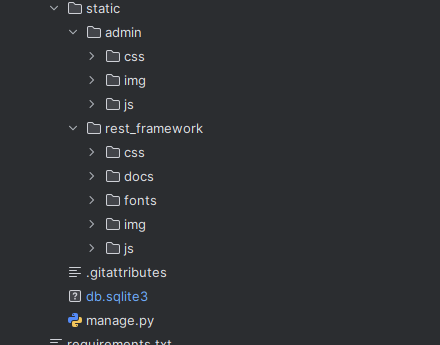
5. E-commerce платформы: За счёт встроенной поддержки безопасности и управляемых транзакций.

Как инструмент, Django поддерживает широкие возможности для расширения и масштабирования, что делает его подходящим решением для большинства веб-проектов. Если ваш проект подразумевает частое взаимодействие с пользователем, нацелен на обработку большого объёма данных или требует мощной административной панели — Django станет отличным выбором.

**Приложение 1 Файловая структура проекта**







**Приложение 2 Список необходимых библиотек**

asgiref==3.8.1

beautifulsoup4==4.12.3

Django==5.0.4

django-bootstrap4==24.4

djangorestframework==3.15.2

numpy==2.1.1

opencv-python==4.10.0.84

pillow==10.4.0

soupsieve==2.6

sqlparse==0.5.1

typing\_extensions==4.12.2

tzdata==2024.2

**Приложение 3 Список использованных источников**

<https://github.com/chuanqi305/MobileNet-SSD/tree/master>

<https://pjreddie.com/darknet/yolo/>

Дронов В.А.Django 4. Практика создания веб-сайтов на Python. 2023

ChatGPT-4o

Giga.chat