

The background features a blue gradient with a complex pattern of white lines and shapes. On the left side, there are circuit-like traces and nodes. On the right side, there is a grid of hexagons, some of which are filled with a lighter blue color, creating a honeycomb-like structure.

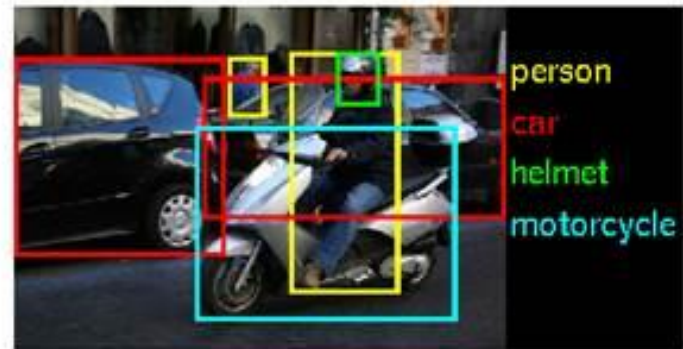
ДЕТЕКТИРОВАН ИЕ ОБЪЕКТОВ

ПЛАН

- **Задача детектирования**
- Демонстрация Tensorflow Object Detection API
- Создание своей модели детектирования
- Демонстрация обучения модели
- Детектирование объектов на видео
- Пример детектирования движущихся объектов, с помощью ImageAI

Задача детектирования

- Требуется определить местоположение всех объектов заданного класса на изображении.
- Класс объектов:
 - Конкретный объект
 - Произвольный регистрационный знак (номер) автомобиля
 - Лицо произвольного человека
 - Произвольный человек
 - ...
- Местоположение:
 - Пиксели, принадлежащие изображению объекта
 - Контур объекта
 - Область изображения, содержащая объект

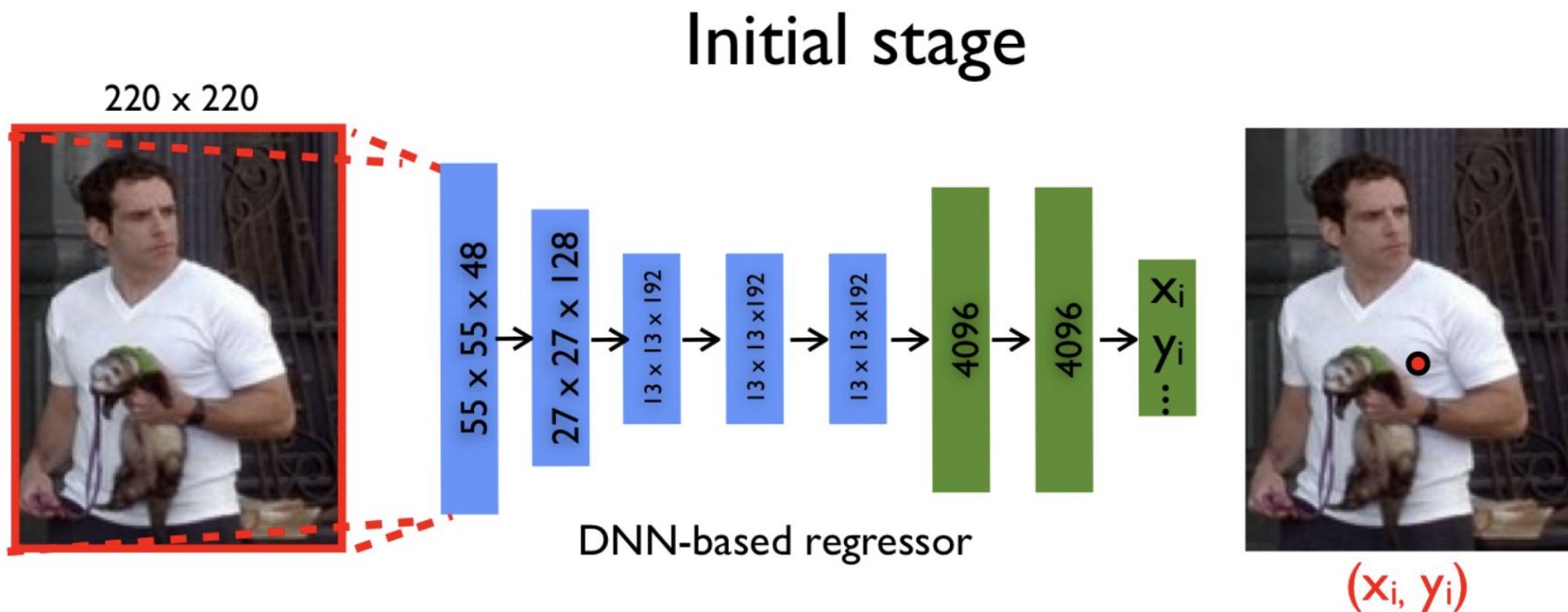


Сложность задачи детектирования

- Сложность определяется:
 - изменчивостью изображений объектов рассматриваемого класса
 - Положение
 - Ракурс
 - Освещение
 - Цвет
 - Форма
 - Частичное перекрытие другими объектами
 - Разнообразием окружающих объектов

Детектирование, как задача регрессии

- На вход нейронной сети подаются изображения
- Выходом является предсказание координат точек, ограничивающих объект



Инструменты применяемые для детектирования

- Tensorflow Object Detection API

https://github.com/tensorflow/models/tree/master/research/object_detection

- YOLO: Real-Time Object Detection

<https://github.com/pjreddie/darknet/wiki/YOLO:-Real-Time-Object-Detection>

- ImageAI

<https://imageai.readthedocs.io/en/latest/>

Демонстрация применения Tensorflow Object Detection API

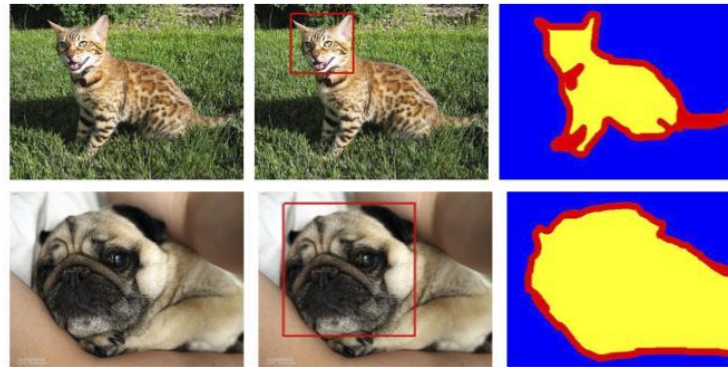
ПЛАН

- Задача детектирования
- Демонстрация Tensorflow Object Detection API
- **Создание своей модели детектирования**
- Демонстрация обучения модели
- Детектирование объектов на видео
- Пример детектирования движущихся объектов, с помощью ImageAI

Примеры наборов данных для обучения

- The Oxford-IIIT Pet Dataset

<http://www.robots.ox.ac.uk/~vgg/data/pets/>



- Pascal VOC Dataset

<https://pjreddie.com/projects/pascal-voc-dataset-mirror/>

Подготовка данных для обучения

- ❑ Инструменты для разметки наборов данных
 - Label Image <https://github.com/tzutalin/labelImg>
 - Open CV Computer Vision Annotation Tool (CVAT) <https://github.com/opencv/cvat>
 - Yolo_Label https://github.com/developer0hye/Yolo_Label

Демонстрация
обучения детектора
объектов на основе
предварительно
обученной модели

ПЛАН

- Задача детектирования
- Демонстрация Tensorflow Object Detection API
- Создание своей модели детектирования
- Демонстрация обучения модели
- **Детектирование объектов на видео**
- Пример детектирования движущихся объектов, с помощью ImageAI

Видео

- Видео – это набор кадров
- Видео можно представить, как тензор с размерностью
(k, h, w)

где k – количество кадров в видео потоке

h – высота кадра

w – ширина кадра

- Значит задачу анализа видео-потока можно свести к задаче анализа изображений

Видео

- Видео – это N кадров в секунду
- Требует большего количества ресурсов
- Предполагает анализ изображения в динамике, а не в статике

Видео

Можно применить тот же набор инструментов, что и для изображений, но с использованием некоторого нового функционала:

- Tensorflow Object Detection API
- YOLO: Real-Time Object Detection
- ImageAI
- OpenCV

Демонстрация детектирования объектов на видео

**СПАСИБО
ЗА ВНИМАНИЕ!**