

Компьютерное зрение

Современные подходы к решению задач компьютерного зрения от алгоритмов до внедрения в продакшн в разных средах

Длительность курса: 106 академических часов

1 От основ к современным архитектурам

1 Компьютерное зрение: задачи, инструменты и программа курса

познакомиться с программой курса;
обсудить организационные вопросы;
познакомиться с инструментами, с которыми будем работать на протяжении всего курса.

Домашние задания

1 Установка и настройка рабочего окружения

Цель: Собрать свой docker контейнер для работы с моделями глубокого обучения.

Необходимо установить и настроить рабочее окружение:

- 1) Установить и настроить Docker.
- 2) Если используете NVIDIA GPU, то также установить nvidia-docker.
- 3) Собрать и запустить контейнер с Ubuntu 18.04 (или выше).
- 4) Развернуть в контейнере PyTorch, TensorFlow, JupyterLab, OpenCV.
- 5) В качестве результата выслать (через личный кабинет) скриншот или лог, который будет содержать версию PyTorch и TensorFlow.
- 6) *Получите дополнительный балл, если в Google Colab запросите GPU (также нужно выслать скрин с версиями PyTorch / TensorFlow).

Пример Dockerfile (для GPU) и Dockerfile_cpu (для CPU) можно найти в материалах к лекции.

2 Сверточные

рассмотреть сверточную нейронную сеть;

**нейронные сети.
Операции свертки,
транспонированной
свертки, пуллинг**

рассмотреть свертку, деконволюцию, субдискретизацию;
Dropout и BatchNorm для сверток.

**3 Эволюция
сверточных сетей:
AlexNet-
>EfficientNet**

выбрать подходящую к задаче архитектуру сверточной
нейронной сети.

Домашние задания

1 Сравнение моделей

Цель: Сделать сравнительный анализ сверточных моделей
для задачи классификации

Посчитать метрики качества на тестовых примерах и
сделать сводную таблицу.
+ Бонусные задания -- реализовать forward свертку,
сделать демо с mobileNet.

Материалы ДЗ прикреплены либо можете скачать их с
colab:
https://colab.research.google.com/drive/1rr1Xq2QGbXs4Epi8__sSJP5jr4Wr7Ysv?usp=sharing

**4 Подготовка и
аугментация
данных**

подготовить данные для обучения под конкретную модель и
задачу;
научиться аргументировать данные для улучшения качества
модели и снижения переобучения.

Домашние задания

1 Датасет

Цель: Собрать и подготовить свой собственный датасет,
можно в соответствии с темой будущего проекта.
Аугментировать данные техниками из занятия.

Примеры:

1. сделать 100 фото предметов на одном/нескольких
фонах, при одном/разном освещении и т.д.
2. снять видео с домашним животным и разметить (кот
лежит, кот бежит, кот ест)

1. Взять предобученную модель

2. Применить несколько стратегий аугментации данных
(alumentations, torchvision или еще что на ваш вкус)

3. Собрать результаты успешности стратегий в сводную
табличку

Большим плюсом будет применение Pytorch Lightning для
упорядочивания вашего проекта, но это не обязательное
требование.

Домашние задания

- 1 Классифицировать игроков в каждом кадре по двум командам на основе цвета их формы

Цель: Скачайте архив с данными <https://disk.yandex.ru/d/NMIVHytktlzbEw> и распакуйте. Внутри архива лежит папка frames и файл bboxes.json. Файл bboxes.json содержит bounding boxes игроков двух команд, а в папке frames расположены соответствующие им изображения (кадры). Всего дано 100 кадров. Каждому кадру соответствует 10 bounding boxes игроков.

Структура файла bboxes.json следующая:

```
frame_n: {
  player_1: {
    'box': [x,y,w,h],
    'team': int
  }
  ....
  player_10: { .... }
}
....
frame_n+k: { .... }
```

Здесь frame_n - номер кадра (ему соответствует файл с изображением в папке frames, всего 100),
 player_1 - id игрока в текущем кадре,
 'box': [x,y,w,h] - bounding box соответствующего игрока (координаты нормализованы от 0 до 1, поэтому их нужно домножить на размеры изображения, чтобы перевести их в целые координаты фрейма),
 'team': int - id команды к которой принадлежит игрок в данном кадре (может принимать значения 0 или 1). Таким образом, id игроков и id команд имеют смысл только в пределах одного кадра и не связаны с другими кадрами. Т.е. в разных кадрах одному и тому же игроку / команде может соответствовать разные id.

- 1) Классифицируйте игроков, используя в качестве вектора признаков игрока (features) средний цвет в пространстве RGB, который берется из области изображения, покрытой bounding box этого игрока (т.е. вектор признаков будет состоять из 3 элементов [r,g,b].
- 2) Используйте в качестве вектора признаков игроков их средний цвет в пространстве HSV (аналогично вектор признаков [h,s,v]).
- 3) Используйте в качестве вектора признаков гистограмму в пространстве RGB, а не средний цвет. Число бинов определите самостоятельно.
- 4) Используйте в качестве вектора признаков гистограмму в пространстве HSV.

В качестве классификатора используйте кластеризацию (например, алгоритм k-means с двумя кластерами).

датасеты и модели в PyTorch на примере Fine-tuning

сетей для их дообучения под собственные датасеты с использованием фреймворка PyTorch.

Домашние задания

1 Катастрофическое забывание

Цель: Проверить влияние fine-tuning на исходную модель

1. Проверить качество исходной модели на ImageNette
2. Используя код с лекции дообучить модель решать датасет CIFAR10
3. Вернуть оригинальный последний слой модели и проверить качество на ImageNet
4. Дообучить только последний слой на ImageNette и проверить удалось ли добиться исходного качества.

7 Object detection 1. Постановка задачи, метрики, данные, R-CNN

спроектировать и научиться обучать модели типа RCNN для решения задачи детектирования объектов.

Домашние задания

1 Object Detection

Цель: Цель: получить практическое представление о работе детекторов объектов, а также метриками для оценки их качества .

1) Скачать датасет:

- Зайти на <https://cocodataset.org/#download>
- Скачать валидационную подвыборку изображений: [2017 Val images](<http://images.cocodataset.org/zips/val2017.zip>)
- Скачать архив с аннотациями: [2017 Train/Val annotations](http://images.cocodataset.org/annotations/annotations_trainval2017.zip)
- Из архива с аннотациями извлечь файл *instances_val2017.json*
- Взять 200 первых изображений и найти к ним все аннотации

2) Прогнать 200 изображений через object detection модель (Detectron2 или YOLOv3/v4/v5).

3) Посчитать IoU предсказанных боксов с ground truth для 200 изображений

4) Посчитать Precision и Recall используя IoU с разными порогами (0.25, 0.5, 0.75, 0.95)

8 TensorRT и инференс на сервере

multiGPU-обучение, подбор гиперпараметров; TensorRT, ONNX, инференс; TFLite.

2 Детекция, трекинг, классификация

1	Landmarks: Facial landmarks: PFLD, stacked hourglass networks(?), Deep Alignment Networks (DAN),	<p>human 2D pose estimation; facial landmarks detection; other landmarks detection.</p> <p>Домашние задания</p> <p>1 Использование Facial Landmarks в качестве Feature Estimator</p> <p>Цель: Создать свой пайплайн Facial Recognition.</p> <ol style="list-style-type: none">1. Выберите Facial Detector по вкусу.2. Выполните Face Alignment.3. На это натравите Facial Landmarks Detector по выбору.4. На этом обучите классификатор на предпочитаемом датасете.
2	Object detection 2. Mask-RCNN, YOLO, RetinaNet	<p>научиться применять альтернативные архитектуры нейронных сетей для решения задачи детекции объектов.</p>
3	Стандартные датасеты и модели в TensorFlow на примере подхода Transfer Learning	<p>научиться использовать готовые модели сверточных нейронных сетей для решения новых задач компьютерного зрения с использованием фреймворка Tensorflow.</p> <p>Домашние задания</p> <p>1 Transfer Learning</p> <p>Цель: Соединим материалы занятий Fine Tuning и Transfer Learning:</p> <ul style="list-style-type: none">* Соберем свой небольшой датасет и проведем несколько экспериментов* Выполним transfer learning fine-tuning нейросети на новой задаче и сравним качество модели на старой задаче до и после обучения новой задаче. <p>См. инструкции в ноутбуке (прикреплен в материалах в преподавательской).</p>
4	Pose estimation	<p>deep pose, convolutional pose machines, stacked hourglass networks, simple baselines for human pose estimation and tracking, работа с 3D pose estimation, unsupervised-методы.</p>
5	Object tracking	<p>научиться применять нейронные сети для решения задачи отслеживания объектов на видео.</p>
6	Face recognition	<p>рассмотреть принципы работы двух основных подходов по распознаванию лиц.</p>

Домашние задания

1 Оптимизация работы с Olivetti

Цель: Цель этого домашнего задания попробовать использование разных классификаторов в качестве Feature Matching.

Взять код с занятия.

Добавить свою Сеть/Модель в конец.

Обучить свой классификатор на PCA датасете.

3 Сегментация, генеративные модели, работа с 3D и видео

1	Сегментация + 3D-сегментация	научиться применять модели типа U-net для задач сегментации изображений.
2	Работаем с 3D сценами. PointNet	<p>научиться решать классические задачи компьютерного зрения на трехмерных объектах.</p> <p>Домашние задания</p> <p>1 Разные способы применения PointNet</p> <p>Цель: После выполнения вы узнаете, как еще можно применять PointNet, кроме задач классификации и сегментации изображений.</p> <p>(так же написано в ноутбуке)</p> <p># Домашняя работа</p> <p>0. Обучите PointNet</p> <p>1. Напишите функцию, которая находит набор критических точек для объекта, сделайте визуализацию нескольких примеров (определение критической точки - раздел 4.3 в [статье](https://arxiv.org/pdf/1612.00593.pdf))</p> <p>2. Напишите функцию, которая ищет похожие объекты по вектору global features (см. раздел E дополнительных материалов к [статье](https://arxiv.org/pdf/1612.00593.pdf)).</p> <p>3. Напишите функцию, которая выполняет операцию shape correspondence для двух объектов одного класса по набору критических точек (см. раздел E дополнительных материалов к [статье](https://arxiv.org/pdf/1612.00593.pdf))</p>
3	Self-driving / Autonomous Vehicle	рассмотреть Self-driving; рассмотреть autonomous Vehicle.
4	Автокодировщики	обсудить, что такое Автокодировщик; обсудить для чего нужны Автокодировщики и какими они бывают; попрактиковаться.
5	Action recognition и 3d для видео	рассмотреть базовые концепции Action Recognition / Detection; поговорить о видеодатасетах для Action Recognition; обсудить различные подходы в задачах обработки видео; рассмотреть как работает 3D Convolution Operator; рассмотреть архитектуры моделей C3D и I3D.
6	GANs 1. Фреймворк, условная	научиться решать задачу порождения изображений и атаки на обученные модели нейронных сетей.

7

Semi-supervised learning

рассмотреть semi-supervised learning.

8

GANs 2. Обзор архитектур

расширить кругозор в области состязательных генеративных моделей и получить практические навыки их применения

4 Проектная работа

- | | | |
|---|---|---|
| 1 | Выбор темы и организация проектной работы | выбрать и обсудить тему проектной работы;
спланировать работу над проектом;
ознакомиться с регламентом работы над проектом. |
| <hr/> | | |
| 2 | Консультация по проектам и домашним заданиям | получить ответы на вопросы по проекту, ДЗ и по курсу. |
| <hr/> | | |
| 3 | Защита проектных работ | защитить проект и получить рекомендации экспертов.

Домашние задания |
| <div>1 Проектная работа</div> <div>Цель: В этом ДЗ вы сделаете и защитите свой проект.</div> <div><div>1. Утвердить тему проектной работы</div><div>2. Сделать проект</div><div>3. Защитить проект</div><div>4. Отправить ссылку на проект в чат с преподавателем</div></div> | | |