Министерство образования и науки Российской Федерации

Федеральное государственное автономное образовательное

учреждение высшего образования

«Южно-Уральский государственный университет»

(Национальный исследовательский университет)

Высшая школа электроники и компьютерных наук

Кафедра системного программирования

ОТЧЕТ

по практической работе №3

## Работа с видео и изображениями (YOLOv5, Detectron2)

|  |  |  |  |
| --- | --- | --- | --- |
|  | | | |
|  | | | |
|  |  | Руководитель: | |
|  |  | | \_\_\_\_\_\_\_\_\_\_\_К.Ю. Никольская  «\_\_\_\_»\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ 2020 г.  Автор работы  студент группы КЭ-220  \_\_\_\_\_\_\_\_\_\_\_\_А.В.Витомсков  «\_\_\_\_»\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ 2020 г. |

Челябинск 2020

1. Развернуть у себя YOLOv5. Необходимо использовать официальный репозиторий (https://github.com/ultralytics/yolov5).

YOLO - Это эффективный и быстрый алгоритм обнаружения объектов, который лучше всего подходит для задач обнаружения объектов в реальном времени.

Устанавливаем pytorch.

conda install -c anaconda mkl

Библиотека Intel Math Kernel Library (Intel MKL) – это математическая библиотека, которая содержит большой набор математических функций и может использоваться в параллельных приложениях для ускорения процесса счета и достижения максимальной производительности.

conda install -c pytorch pytorch torchvision

Пакет Torchvision состоит из популярных наборов данных, архитектур моделей и шаблонов для компьютерного зрения.

CUDA v11.0 устанавливается вместе с pytorch.

conda install -c anaconda cudatoolkit version==10.2

Установка yolo5:

git clone <https://github.com/ultralytics/yolov5.git>

Если вы используете windows, отредактируйте requirements.txt файл в папке yolov5 и заменить строку «pycocotools» с линией «pycocotools -Windows» . Для Linux это не нужно.

pip install -U -r requirements.txt

Запуск yol\_v5:

Рабочий каталог C:\Users\aleks\anaconda3\yolo5\yolov5

cd /d “C:\Users\aleks\anaconda3\yolo5\yolov5”

**Запуск преобразования выполняется командой:**

python detect.py --source ./inference/images/ --weights yolov5s.pt --conf 0.4

Здесь:

-source – папка с анализируемыми изображениями,

-weight – расположение файла с весами (yolov5s.pt обучены на наборе данных COCO),

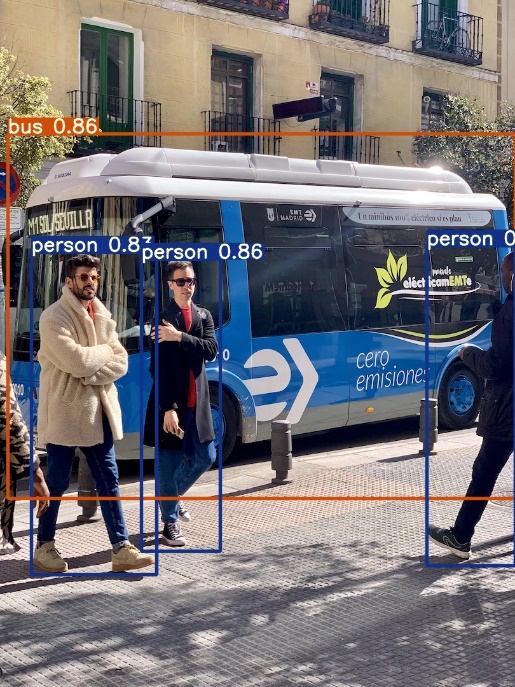
-conf - если объект на изображениях совпадает с каким-либо классом в наборе данных COCO с достоверностью больше или равной conf, то он детектируется.

MS COCO (англ. Common Objects in Context) — большой набор изображений. Состоит из более чем 330000 изображений (220000 — размеченных), с более чем 1.5 милионов объектов на них. Все объекты находятся в их естественном окружении (контексте). Изображения, как правило, содержат объекты разных классов (только 10% имеют единственный класс). Все изображения сопровождаются аннотациями, хранящихся в json формате.

Тестирование.

python detect.py --source ./data/images/ --weights yolov5s.pt --conf 0.4

файлы в /runs/detect/:

2. Установка detectron2.

Конфигурация та же что и для yolov5, только требуется numpy v1.19.2.

conda install --force-reinstall numpy=1.19.2

git clone <https://github.com/DGMaxime/detectron2-windows.git>

cd “C:\Users\aleks\anaconda3\detectron2\detectron2-windows”

если появляется ошибка с numpy, просто удалить папку numpy-1.19.2.dist-info.

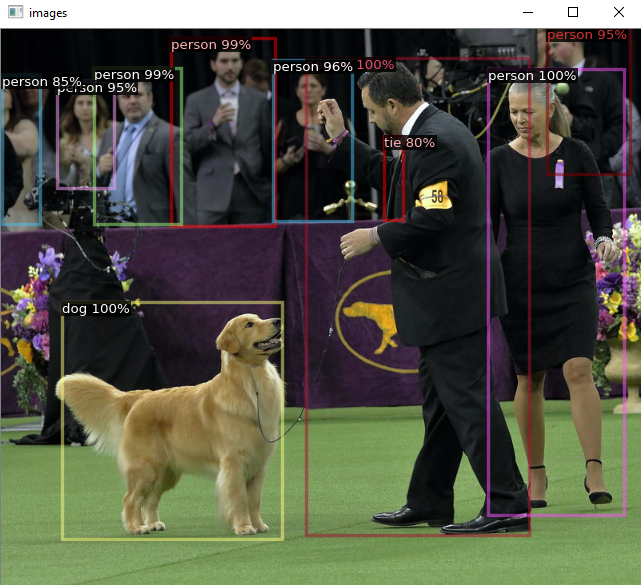
pip install opencv-python

Тестирование установки:

python tests/test\_windows\_install.py

Перед этим в test\_windows\_install.py ставим

cfg.MODEL.DEVICE = "cpu"



3. Возьмите любой размеченный набор данных и обучите на развернутых архитектурах.

Я выбрал набор данных SCUT-HEAD dataset с изображениями аудитории. <https://github.com/HCIILAB/SCUT-HEAD-Dataset-Release>

PartA SCUT-HEAD включает 2000 изображений, взятых из видеоматериалов с мониторов учебных аудиторий в университете, с размеченными 67321 объектами. 1500 изображений PartA предназначены для обучения и 500 - для тестирования.

Результаты тестирования.

Несмотря на то что мы обучаем модель на собственном наборе данных, выгодно не обучать её с чистого листа, а применить «перенос обучения». То есть использовать в качестве отправной точки веса от другой, уже обученной модели. Здесь работает следующая аналогия: чтобы забраться на конкретное место горы быстрее, мы идём не случайным образом, а по проложенной ранее тропинке.

Чтобы обучить наш детектор объектов, нам нужно контролировать его обучение с помощью аннотаций ограничивающего прямоугольника. Мы рисуем рамку вокруг каждого объекта, который должен видеть детектор, и помечаем каждое поле классом объекта, который детектор должен предсказывать.